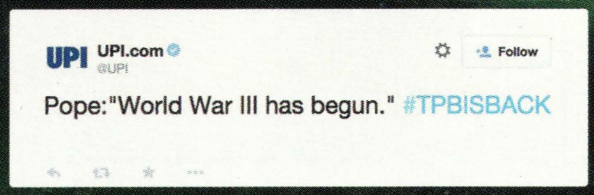
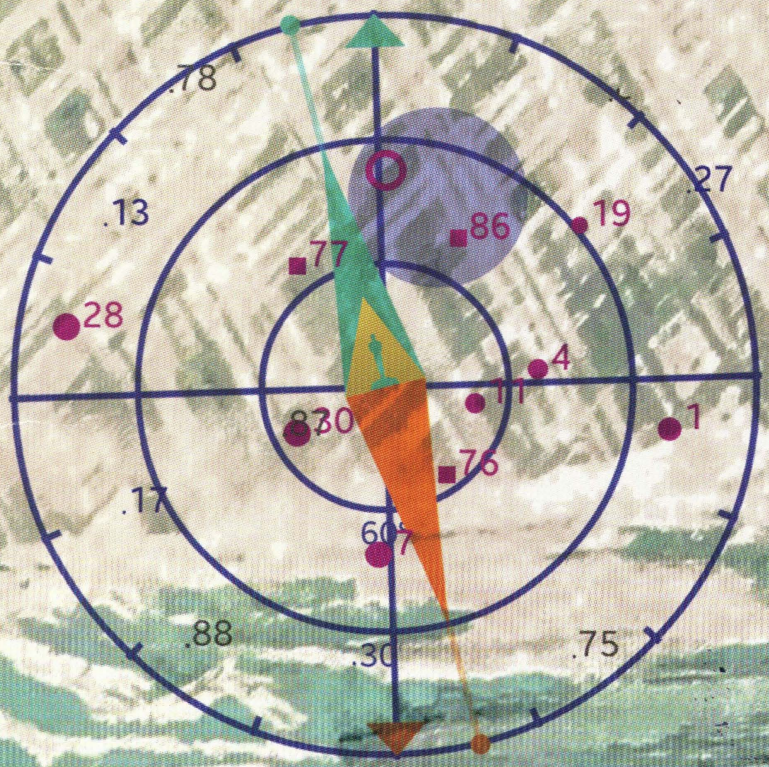
Volume Thirty-Two, Number One!
Spring 2015, $6.95 US, $7.50 CAN

2600

The Hacker Quarterly





Lattitude
78.252785N

Longitude
15.409617E

Island Payphones

Taiwan. Seen in the Maokong District of Tapei, this payphone, like all in this city, is also a free Wi-Fi hotspot.

Photo by Matt Ranostay

Tahiti. It may look like a painting, but we can assure you this phone is very real and functional.

Photo by nfltr8

Japan. This phone was found on the country's largest island of Honshu on the road side of Nagano Prefecture and it's still in pristine condition.

Photo by RayD

Fiji. Discovered on Taveuni Island, this phone looks like it's prepared to attack anyone who offends it in any way.

Photo by Daniel Eather

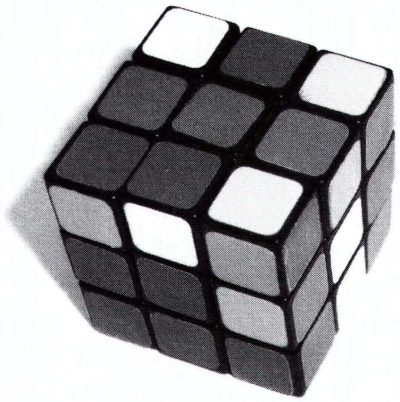
Got foreign payphone photos for us? Email them to **payphones@2600.com**. Use the highest quality settings on your digital camera! (Do not send us links as photos must be previously unpublished.) (More photos on inside back cover)

[Array]

Nous Défions Tout	4
So, You Want to Be a Darknet Drug Lord....	6
Out of the Box Survival - A Guide to PowerShell Basics	10
TELECOM INFORMER	13
Brazil's Electronic Voting Booth	15
Evolution of a Hack	17
Bleeper - Downloading Full-length Preview MP3s from bleep.com	21
The Enterprise, the Subverted PKI Trust Relationship, and You	23
HACKER PERSPECTIVE	26
WYSE Moves	30
Office Talk or Social Engineering?	31
Archiving ComiXology	33
LETTERS	34
McAfee Family Protection - Epic Fail!	48
Abusing the Past	50
Hacking the HandLink Gateway	51
EFFECTING DIGITAL FREEDOM	52
Ohio Prison IT Security from the Inside	54
Hacking For Knowledge	55
Linux Containers for Event Training	57
Are Smart Meters the End-All?	59
HACKER HAPPENINGS	61
MARKETPLACE	62
MEETINGS	66

LLAP

Nous Défions Tout



Hackers continue to be catalysts for change, scapegoats for every imaginable problem, and an unending source of ratings for news programs and inspiration for movie plots. Lately, though, this atmosphere has really been in high gear.

It doesn't help lessen the volume when every time a computer system is compromised or falls apart due to its own flimsiness that hackers are the ones deemed responsible. Hackers even get blamed when such scenarios are averted ("were it not for the investigation by this intrepid reporter, hackers *could* have been able to steal your identity..."). We need to all face the fact that there are lots of people out there with agendas who also happen to know how to use computers. While hackers can figure out tricks and security vulnerabilities (as well as figure out how to fix them), it doesn't take that same ingenuity to simply apply them en masse to target systems. It's simply an end user attack using hacker tools. Most anyone can do this.

That's why we thought it amusing that last year's attack on Sony was being attributed to an army of hackers from North Korea. First, as we've seen demonstrated often, hackers don't work well in armies. They tend to act as individuals and question all that is around them, which is what makes them good hackers in the first place. Hackers aren't particularly good at following orders, hence the large amount of them who wind up in detention at school and otherwise imprisoned elsewhere in life. (North Korea certainly wouldn't be a great environment for independent-minded hackers to thrive, not to mention that the extremely limited bandwidth into the country would make it a trivial task to cut them off.) Sure, there was hacking involved, but not in the way it was being portrayed virtually everywhere.

The security practices at Sony were unsurprisingly the biggest culprit. This is most always the case whenever you see a massive computer breach. We may never know the full story, but it's clear that way too much access was being given to certain users, far too much private data was being stored in an unencrypted form on a system connected to the Internet, and not enough attention was being paid to potential compromises. Supposedly their system had been owned for months before anyone thought to do anything about it. The company had gone data happy - storing everything and anything they could online and having it all connected to the network. What possible reason would there be to have unreleased movies stored online? Clearly, this just wasn't thought through.

Sony is far from alone in this. It's likely the majority of companies in existence today have serious lapses in digital judgment, keeping things online that should be isolated, going overboard with storing personal data, assuming everything is running smoothly without paying attention, etc. That's the real problem, and it's one that affects all of us because oftentimes that data belongs to us. When a bank or credit card company lets one of its massive databases leak onto the net or into someone else's account, it's our privacy that's the victim - without us having any say.

Above all else, though, when such stories hit the mainstream, beware of the spin that's inevitably attached to them. In the case of Sony's *The Interview*, there was a lot going on that escaped scrutiny. (We're referring to the ill-conceived movie that focused on the "hilarious" assassination of North Korean leader Kim Jong-un. To put it into perspective, if such a film were to be made with the same exact plot involving one of *our* leaders, it

would probably be considered an act of terror, so it's hard to believe Sony was that surprised by North Korea's lack of enthusiasm for the project.) For one thing, the initial hacking of Sony's network had no stated connection to the production of this rather controversial film. It was only after pundits theorized that maybe there could be a connection that messages related to the film began to be sent from the alleged perpetrators. Everyone involved - from Sony to the intruders to the media - seemed to be latching onto this perceived issue in order to get themselves more publicity. We had a bit of fun with it ourselves, offering to show the movie on our own website when Sony suddenly decided it was too hot for them to touch. Of course, in the end they relented (or went along with their original plan, depending on which conspiracy you believe) and *The Interview* wound up doing quite well when it otherwise would have been largely ignored.

But another important point was illustrated with all of this, regardless of how we may have been manipulated. No matter how bad or offensive a particular statement, idea, or presentation is, being told you're not allowed to see, create, or talk about such a thing is far worse. In fact, nothing makes such a thing come to life more than turning it into forbidden speech.

This is nothing new to the hacker world. Whenever we're confronted with something we're "not allowed to know," we move mountains to learn all about it anyway. That is the life blood of this publication. Such a mentality extends to the Internet, where censorship is said to be thought of as a network problem that can be routed around. While *The Interview* got consistently poor reviews, it did well because we were ostensibly told we weren't allowed to see it. Brilliant marketing or the spirit of freedom, perhaps a mix of the two.

We've seen a similar - and far more serious - example of resistance to forbidden expression with the recent tragedy in Paris. Being told one cannot illustrate or disrespect an icon of a religion (in this case, the Prophet Mohammed) is anathema to anyone who truly believes in freedom of speech. In fact, we focused on this very idea in our Spring 1989 issue (detailed more thoroughly in Volume 6 of *The Hacker Digest*) when author Salman Rushdie had a bounty put on his head by Iran's Ayatollah Khomeini for his writings on Islam. It was no wonder the hacker community at the time took note of this incident. We've never reacted well to being told what we can and cannot say.

In the *Charlie Hebdo* case, the irony is particularly biting since the victims of this massacre were those who probably felt most passionately about protecting the rights of *anyone* under assault, whether it be for religious, ideological, or political reasons. They were certainly no friends of the ugly nationalism and religious intolerance that has been springing up in France and other countries, whether as a reaction to this kind of atrocity or because it never really disappeared in the first place. The journalists who were mowed down on that dark day in January represented that part of us dedicated to rebelling against any power attempting to control us through our speech. It's not a Western value or something that is alien to anyone on earth. It's a human trait. If you tell someone they're not allowed to say something, the very first thing an individual will do is say it, whether out loud or to themselves. It has nothing to do with whether or not they believe it and everything to do with their right to process their own thoughts and reach their own conclusions.

It's easy to point to a case like this because of its magnitude and the perceived culture clash which is being exploited by all sides. But one doesn't have to look far to find an unending supply of other instances of journalists and common citizens being victimized because they asked the wrong question or made the wrong statement. You would be hard pressed to find a regime with the high moral standing to condemn what happened here without their looking like complete hypocrites. Governments, corporations, religions, institutions of all sorts are filled with conflicted statements and positions that simply make it impossible for them to judge their counterparts with any true legitimacy. As individuals, though, we have more power to confront our contradictions, to rethink our philosophies, and to challenge anything we're expected to accept without question.

This is what being a hacker has always been about. We don't fit into the agendas of large organizations and we don't take orders. We're there to challenge the status quo, to mess with the system, ask a million questions, and always try and come up with something different and better. And if you look back throughout history, you'll see that such challenges never come cheap.

SO, YOU WANT TO BE A DARKNET DRUG LORD....

by nachash
nachash@observers.net

The advice in this article can be adapted to suit the needs of other hidden services, including ones which are legal in your jurisdiction. The threat model in mind is that of a drug market. The tone is that of a grandfather who is always annoyingly right, who can't help but give a stream-of-consciousness schooling to some whippersnapper about the way the world works. If this article inspires you to go on a crime spree and you get caught, don't come crying to me about it.

You've decided that you're bored with your cookie-cutter life of working at a no-name startup, getting paid in stock options and empty promises. You want a taste of the good life. Good for you, kid. I used to run a fairly popular hidden service (DOXBIN) that was seized by the FBI after three and a half years of spreading continuous butthurt, then subsequently repossessed from the feds. Because I managed to not get raided, I'm one of the few qualified to instruct others on hidden services and security, simply because I have more real-world experience operating hidden services than the average Tor user. In other words, very little of this advice is of the armchair variety, as you'll often find in abundance on the Internet. But enough about me. Let's talk about your future as an Internet drug lord.

Legal/Political

First things first, you need to cover the legal, historical, and political angles. Read up on various drug kingpins and cartels from the 20th century. Learn everything you can about how they rose and fell (you can safety ignore all the parts about intelligence agencies backing one drug cartel over another, because that's not going to happen to you). Once you've got a good command of that, read everything you can about busted drug market operators and branch out into cybercrime investigations as well. It wouldn't hurt to make yourself familiar with law enforcement and intelligence agency tactics either. You'll find that virtually all drug kingpins either get murdered or go to prison. Let those lessons sink in, then find a good drug lawyer and make plans for being able to pay them when The Man seizes everything you own. While you're dreaming big about making fat stacks of fake Internet money, do some research on Mutual Legal Assistance Treaties and extradition treaties.

Mutual Legal Assistance Treaties (MLATs) are self-explanatory. Country A will help Country B do whatever it takes to aid a cybercrime investigation should some aspect of the crime bleed over into Country A. Figure out which countries don't provide legal assistance to your country in these cases, then find hosting services that are based there. You'll shorten this list by determining which hosts allow Tor, or at least don't explicitly forbid it in their Terms of Service (you don't care about exit bandwidth. You just want relays. Remember this for later in the article). Last but not least, sort out which hosts accept payment options that don't make you sweat bullets over the fact that the NSA has been monitoring global financial transactions since at least the 1970s. You will want to avoid any host that advertises itself as bulletproof - they'll probably kit your box and siphon everything of value, in addition to overcharging you for the privilege of running on older hardware - and any host which sells a cheap VPS and promises to guarantee your privacy.

Extradition treaties mean that if you're in Country A and do something that makes Country B want to prosecute you, Country A is most likely going to give you a one way ticket to Country B. If or when your box gets seized and you know the heat is on, you're going to want to beat it to a place that won't send you back, where you will presumably live out the rest of your days. Just make sure you've made enough money to grease all the right palms in your new life, or the road ahead may be extremely bumpy. If you're smart, you'll permanently move to this country well before you have any trouble with law enforcement.

One last thing before moving on: Don't be so stupid as to attempt to hire a hitman to kill anyone. Murder-related charges have no statute of limitations, which means you won't get to write a tell-all book about what a sly bastard you are when this wild ride is a distant memory. If you've reached a point in your new career where murdering people makes sense, it's time to walk away. Don't get corrupted like Dread Pirate Roberts.

Technical

This section tries to be as operating system independent as possible. You'll want to consult

the documentation of your OS for specifics. The technical side of running a hidden service and not getting owned by cops is a lot harder than just installing stuff and crossing your fingers. The recommendations in this section *will not* protect you from zero-days in the wild, but should help somewhat with damage control. Remember, if they want to own your hidden service, it will probably happen eventually.

Before you even think about installing BitWasp and Tor, you need to really understand how Tor works. Go to `freehaven.net` and read the white papers until your eyes glaze over, then continue reading until you're out of papers to read. Pay particular attention to the hidden service papers. If you feel like you didn't understand something, come back to that paper again when you have more knowledge. A lot of the papers explain some of the same concepts with slight differences in the intros. Don't skim over them, because you might read someone's rewording that will clarify an idea for you. Check back with Free Haven regularly. Once you're up to speed, a good next step is to keep up with The Tor Project's mailing lists.[1]

While you're doing all of this reading, it's (mostly) safe to go ahead and install Tor on a box on your local network, purely for experimentation. Keep in mind that the NSA will start scooping up all of your packets simply because you visited `torproject.org`. That means don't post code questions related to your drug market on Stack Exchange if you want to avoid giving The Man morsels he can use for parallel construction. Once you've gotten hidden services working for http and ssh, you're going to take the first baby step towards evading casual discovery: Bind your hidden services to localhost and restart them.

The next step in your journey towards changing the drug business forever is to grab the transparent proxying firewall rules for your operating system to make sure they work.[2] They will guard against attacks that cause your box to send packets to a box the attacker controls, which is useful in thwarting attempts to get the box IP. You may wish to have a setup similar to an anonymous middle box, preferably without public IPs where possible, so if your application gets rooted, Tor isn't affected.

Speaking of applications, do everything you can to ensure that the application code you use to power your hidden service isn't made of Swiss cheese and used Band-Aids. To protect against other types of attacks, you will want to identify any pre-compiled software that your users will touch and compile it yourself with hardening-wrapper or its equivalent, plus any custom flags you want to use. If you keep vulnerabilities from the application and server to a minimum, your biggest worries will be Tor-related.

You will only connect to your production box via a hidden service. It's a good idea to get into that habit early. The only time deviating from this pattern is acceptable is when you have to upgrade Tor, at which time you'll want to have a script ready that drops your firewall rules and unbinds SSH from localhost just long enough for you to login, do the upgrade, re-apply the firewall rules and bind SSH to localhost again. If you're not ready to deal with the latency, you're not ready to do any of this. Don't forget to transparently proxy the machine you use too, so you don't slip up by mistake.

On the subject of the machine, you need to automate the process of both setting up your hidden service and of destroying it. Proactively change servers every few months, in order to frustrate law enforcement attempts to locate and seize your site. Your creation script should install everything your site needs as well as all configuration files. Your clean-up script needs to destroy all evidence, preferably with a tool like srm.

Regarding time-related issues: Always select either UTC or a time zone that doesn't match the box's location. You will also do this to the box you use to interact with your hidden service every day. If you read the white papers, you will probably note a recurring theme of clock skew-related attacks, mostly directed at clients, in some of the older papers. Tor won't even start if the clock skew is off by too much.

If you want to have some fun at the expense of business in the short term, intentionally take your service offline periodically in order to mess up attempts to match your downtime with public information. If you're the kind of person with access to botnets, you could DDoS (Distributed Denial of Service) some provider at the same time on the off chance that someone might connect the dots. This countermeasure will only work on researchers looking at public info, not nation state actors with an ax to grind.

I've saved some of the hardest stuff for the last part of this section. It's hard because you have to make choices and it's unclear which of those choices are the best. It's a bit like a Choose Your Own Adventure book. In that spirit, all I can do is lay out the possibilities in as much of a Herodotus-like way as possible.

One thing you have to consider is whether you want to run your hidden service as a relay

or not. If it's a relay, you'll have extra cover traffic from other innocent Tor users. But if your relay goes down at the same time as your hidden service, it will be far more likely to be noticed. Federal criminal complaints make a big deal of seized hidden services not being relays, but three relays were taken down at around the same time as Operation Onymous, so that's not a guaranteed defense. The choice is yours.

Remember when I said to take note of hosts that don't ban Tor outright? This is the part where you give back to the community in the form of Tor relays or bridges.[3] The feel-good aspects of this move are along the same lines as drug barons who build schools and hospitals, but this is more immediately self-serving. You're going buy several servers to set up strictly as relays or bridges, then configure your hidden service box to use only those relays or bridges to enter the Tor network. Here's where things start to get theoretical.

If an adversary is running a guard node discovery attack - in which an attacker is able to determine the node you're using to enter the Tor network - against your service and you're using your own relays as entry nodes, the damage they can do will be limited to DoS (Denial of Service) if your relays are not linkable to your identity. However, if you're entering the Tor network with bridge nodes, an attacker will probably say "WTF?" at first unless they determine they've found a bridge node. Bridge nodes don't use nearly as much bandwidth as relays because there is not a public list of them, so an intelligence agency would have less traffic to sift through, which makes correlation easier. On the other hand, using bridge nodes also allows you to run obfsproxy[4] on both the bridges and your hidden service. obfsproxy allows you to make Tor traffic appear to be another type of traffic, which is a good defense against non-Five Eyes entities. For example, your hosting provider may decide to monitor for Tor traffic for their own reasons. Just make sure your relays/bridges aren't linkable to you or to each other.

One last thing about guard node discovery attacks: The Naval Research Lab published a paper in July 2014 about the "Sniper Attack,"[5] which in short works like this: The attacker discovers your guard nodes, then uses an amplified DoS trick to exhaust the memory on all of your nodes. The attacker keeps doing this until your hidden service uses guard nodes that they control. Then it's game over. If your hidden service's entry nodes are all specified in your torrc file and they get DoSed, your service will go offline. In this situation, if all of your relays are down, you essentially have an early warning canary that you're being targeted. In other words: This is the best possible time to book your one-way ticket to your chosen non-extradition country. For those of you with a background in writing exploits, this is similar in principle to how stack smashing protection will render some exploits either unable to function or will turn them into a DoS. Personally, I recommend an ever-changing list of relays or bridges. Add a few new ones at a predetermined interval, and gradually let old ones go unpaid.

Operational Security

This section is critical, especially when things start to break down. If everything else goes bad, following this section closely or not could be the difference between freedom and imprisonment.

This is important enough to restate: Transparently proxy your Tor computer. This is a good first line of defense, but it is far from the only way to protect yourself.

Do not contaminate your regular identity with your Onion Land identity. You're an aspiring drug kingpin. Go out and pay cash for another computer. It doesn't have to be the best or most expensive, but it needs to be able to run Linux. For additional safety, don't lord over your new onion empire from your mother's basement, or any location normally associated with you. Leave your phone behind when you head out to manage your enterprise so you aren't tracked by cell towers. Last but not least for this paragraph, don't talk about the same subjects across identities and take countermeasures to alter your writing style.

Don't log any communications, ever. If you get busted and have logs of conversations, the feds will use them to bust other people. Logs are for undercover cops and informants, and have no legitimate use for someone in your position. Keep it in your head or don't keep it at all.

At some point, your enterprise is going to have to take on employees. Pulling a DPR move and demanding to see ID from high-volume sellers and employees will just make most people think you're a fed, which will leave your potential hiring pool full of dumbasses who haven't even tried to think any of this out. It will also make it easier for the feds to arrest your employees after they get done arresting you. If your enterprise is criminal in nature - whether you're selling illegal goods and services or you're in a repressive country that likes to reeducate and/or kill dissidents - an excellent way of flushing out cops

is to force them to get their hands not just dirty, but filthy, as quickly as possible. Don't give them time to get authorization to commit a crime spree. If there's a significant amount of time between when they're given crimes to commit and the commission of those crimes, you need to assume you've got an undercover cop on your hands and disengage. If they commit the crime(s) more or less instantly, you should be fine unless you've got the next Master Splynter on your trail.[6]

Disinformation is critical to your continued freedom. Give barium meat tests to your contacts liberally.[7] It doesn't matter if they realize they're being tested. Make sure that if you're caught making small talk, you inject false details about yourself and your life. You don't want to be like Ernest Lehmitz, a German spy during World War II who sent otherwise boring letters about himself containing hidden writing about ship movements. He got caught because the non-secret portion of his letters gave up various minor personal details the FBI correlated and used to find him after intercepting just 12 letters. Spreading disinformation about yourself takes time, but after a while the tapestry of deceptions will practically weave itself.

Ensure that your communications and data are encrypted in transit and at rest whenever applicable. This means PGP for email and OTR for instant messaging conversations. If you have to give data to someone, encrypt it first. For the Tor-only box you use for interacting with your hidden service, full disk encryption is required. Make a password that's as long and complex as you can remember ("chippy1337" is not an example of a good password). Last but not least, when you're done using your dedicated Tor computer, boot into Memtest86+. Memtest86+ is a tool for checking RAM for errors, but in order to do that it has to write into each address. Doing so essentially erases the contents of the RAM. Turning your computer off isn't good enough.[8] If you're planning to use Tails, it will scrub the RAM for you automatically when you shut down. Once your RAM is clean, remove the power cord and any batteries if you're feeling extra paranoid. The chips will eventually lose any information that is still stored in them, which includes your key. The feds can do a pre-dawn raid if they want, but if you follow this step and refuse to disclose your password, you'll make James Comey cry like a small child.

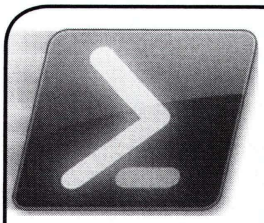
Use fake info when signing up for hosting services. Obfuscate the money trail as much as possible and supply fake billing info. I prefer registering as criminals who are on the run, high government officials, or people I dislike. If your box gets seized and your hosting company coughs up the info, or if a hacking group steals your provider's customer database (it happens more often than you'd think), your hosting information needs to lead to a dead end. All signs in Operation Onymous point to operators being IDed because they used real info to register for their hosting service and then their box got decloaked.

Speaking of money, you're going to have to figure out how to launder your newfound assets, and we're not talking about using a couple of bitcoin laundering services and calling it a day. You also shouldn't go out and buy a Tesla. Living beyond your means is a key red flag that triggers financial and fraud investigations. Remember, money is just another attack vector. Washing ill-gotten gains is a time-honored drug business tradition and one that you would be a fool not to engage in. You can only use your hard-won profits to send `shitexpress.com` packages to people you don't like so many times.

Take-away: If you rely only on Tor to protect yourself, you're going to get owned and people like me are going to laugh at you. Remember that someone out there is always watching, and know when to walk away. Do try to stay safe while breaking the law. In the words of Sam Spade, "Success to crime!"

Sources

1. `https://lists.torproject.org/cgi-bin/mailman/listinfo`
2. `https://trac.torproject.org/projects/tor/wiki/doc/TransparentProxy`
3. `https://www.torproject.org/docs/bridges`
4. `https://www.torproject.org/projects/obfsproxy.html.en`
5. `http://www.nrl.navy.mil/itd/chacs/biblio/sniper-attack-anonymously-deanonymizing-and-disabling-tor-network`
6. `http://www.pcworld.com/article/158005/article.html`
7. `https://en.wikipedia.org/w/index.php?title=Canary_trap&oldid=624932671`
8. `https://freedom-to-tinker.com/blog/felten/new-research-result-cold-boot-attacks-disk-encryption/`



Out of the Box Survival, Part One
A Guide to PowerShell Basics

by Kris Occhipinti - Metalx1000

I am primarily a Linux user. So when I sit down at a computer, I'm used to having development tools at my disposal. On most Linux machines, you are going to have interpreters for Bash, Python, and Perl already installed and ready to go. You will also, in many cases, have compiles for both C and C++ either already installed, or quickly installed, through the use of whichever package manager your distro uses.

This is why I find it so frustrating to sit down at a Windows machine. Microsoft Windows provides you with almost nothing of use when it come to development (or really anything for that matter). Although you can install interpreting tools such as Python, Perl, and even Bash, they aren't already there, out of the box, as they are on a Linux system.

In the past we have had to survive with tools such as batch files and VBS scripts, which are both very limited to say the least. Batch files are not very useful without implementing external tools, which also need to be installed as they are not distributed with Windows. And if you want to install a compiler, they are either overblown in size, giving you way more than a simple compiler, or they tend to be missing library and header files that you require.

It's important to have development tools if you are going to be doing anything of use on a computer beyond simple web surfing or document reading. Whether you are working as IT for a living and just trying to get the system to automate tasks, or if you are up to no good and trying to do something malicious on a system quick and easy, it saves you a lot of headaches when the tools you need are already on the system and you don't have to go looking for them and installing them.

In my search to try and make Windows do something useful out of the box, I was very disappointed. But in recent years, Microsoft has upped their game quite a bit. Since Windows Vista, Microsoft Windows has been packaged with PowerShell, which, as much as I dislike Microsoft (if you couldn't already tell), is pretty powerful.

I would not recommend someone learn to program in PowerShell because they want to learn to program. There are way better tools for creating programs, and I recommend learning a language that has the ability to run on more than one platform (which is pretty much everything non-Microsoft). Definitely don't limit yourself by learning a restrictive language as your first language. But, if you are already familiar with programing but desire the power to be able to sit down at a Windows machine and just start typing and create something useful, PowerShell seems like the best option available at this time. But do remember that this is a no go on anything prior to Windows Vista. So no Windows XP, even though there are plenty of those systems still out there.

When it comes to learning any programming language, there are a handful of things you need to learn off the bat. Once you learn these few basic things, you know 90 percent of what you are going to be doing over and over again. Those basic things are:

- Output to the screen
- Input from the user and storing that input to a variable
- Writing to a file
- Reading from a file
- Sending and retrieving data from the Internet

Beyond these few basic things, the majority of what you will be doing is manipulating the data you get from the user, file, or Internet. Today my goal is to teach you these basics so you can get going with creating your own tools and scripts.

Let's first look at sending output to the screen for the user to see. This, of course, is the classic "Hello World" program. As with most languages this is fairly simple when it comes to printing the words in a terminal.

```
[code]
Write-Host "Hello World"
[/code]
```

PowerShell allows for some more advanced GUIs, but you can also create basic dialog boxes. Here is an example of that.

```
[code]
[System.Reflection.Assembly]::
➥LoadWithPartialName("System.
```

```
➥Windows.Forms")|out-put null
[System.Windows.Forms.
MessageBox]::Show("Hello World!"
, "Welcome")
[/code]
```

I think that is pretty straightforward and doesn't need much explaining. First we load up our forms functions and then create a dialog box. Then you have your main message and a title for the box. Let's now move on to getting input from the user. First we'll look at getting text from the user in a terminal window.

```
[code]
$name = Read-Host 'What is your
➥ name?'
$pass = Read-Host 'What is your
➥ password?' -AsSecureString
[/code]
```

Using the "Read-Host" function, you can post a message and then wait for the user input, while at the same time you can put the input into a variable just as I did here when asking for the user's name. Adding the "-AsSecureString" will hide the user's input as they type, which makes it nice for getting private data such as a password.

Let's look at that same example using the system's built in credential prompt screen.

```
[code]
$cred = $host.ui.promptfor
➥credential('','','','');
$name = $cred.username;
$password = $cred.getnetwork
➥credential().password;
[/code]
```

Or with a little more info added:

```
[code]
$cred = $host.ui.promptfor
➥credential('Failed Authentica
➥tion','',[Environment]::User
➥DomainName + "\" + [Environment
➥]::UserName,[Environment]::User
➥DomainName);
$name = $cred.username;
$password = $cred.getnetwork
➥credential().password;
[/code]
```

This is a little more complex than the "Hello World" GUI, but not by much. Here you can see we are creating a dialog using the "promptforcredential" function. We are giving that dialog a title of "Failed Authentication", making the user believe that something has failed and that they need to re-enter their username and password - which we later place into variables.

Let's once again get input from the user and this time write that data into a file.

```
[code]
$name = Read-Host 'What is your
➥ name?'
$name | out-file ".\name.log"
[/code]
```

Again, this is pretty straightforward. We get the user's name and place it into a variable called "$name" and then we take that data and pipe it into the "out-file" function, which places the user's name into a file called "name.log".

Since we now have data in a file, at some point we might want to be able to retrieve that data. So let's do a simple file read with the "Get-Content" command.

```
[code]
Get-Content ".\name.log"
[/code]
```

This command will just get the data from the file - in this case the user's name - and display it to the screen. If we wanted to store it into a variable for use later in our program, we can do that as well by issuing this command.

```
[code]
$name = Get-Content ".\name.log"
Write-Host $name
[/code]
```

Here we have the "Get-Content" command reading the "name.log", but this time placing all of the data from that file into a variable called "$name". Right after that, we are issuing the command to write that data to the screen.

Seems a little silly in this example, but can be very useful when creating a real script that has purpose.

Lastly, let's play with network connections and access the Internet. A computer these days is pretty much useless without the Internet. If you want to be able to send or retrieve information from a server that you or someone else has set up, you are going to need to know how to script it out. Whether it's as a system admin trying to scan a system and retrieve data remotely, or as an unauthorized user trying to scan a system and retrieve data remotely, the ability to do this with tools that are already on the system is a relief.

Let's look at just downloading a file for now. We can do it in just a few lines of code.

```
[code]
$webclient = New-Object System
➥.Net.WebClient
$webclient.DownloadFile("http://
➥i.ytimg.com/vi/iDpwKiRKmZc/
➥0.jpg","downloaded.jpg")
[/code]
```

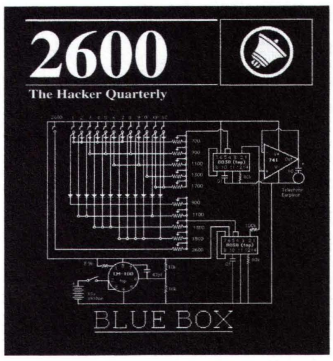
Normally, I would have stored the URL and the file name into variables first, but I wanted to keep this as short as possible for you. We have two lines. The first is creating a new WebClient object. The second is using that object to download a JPEG and save it to a file locally. Here we are downloading an image, but we can download any type of file. It could even be other tools you need for your script. So, even if Powershell doesn't meet all of your needs, you can use it to quickly get all of the tools you do need (although I do suggest using as many built-in tools as possible).

There is much more I want to show you. I want to show you how to put all of this into scripts and work around Microsoft's poor security on how these scripts run. I want to show you how to get PowerShell scripts from a remote server and run them in RAM without touching the hard drive. I also want to show you how to package these script into an EXE file for easy execution. I plan on expanding on these and other PowerShell abilities in future submissions. This is all for now. Enough to get you started playing with PowerShell.

For more programming tips check out: `http://filmsbykris.com`.

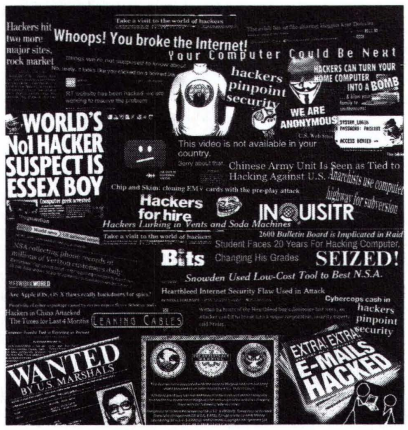
A hacker makes the most out of what they have. They take the technology that is in front of them and change it, recreate it, and repurpose it to solve the problems they face. In the case of Windows OS, tools are limited on a default install. Learn what is available. Use those tools to their fullest and beyond. Become comfortable with them so that you know you can sit down at a machine and know that you can make it do whatever you want without anything but a keyboard. No need to copy or install excess garbage. Keep it light. Like a ninja, go unnoticed, because you are just using the system and its own tools.

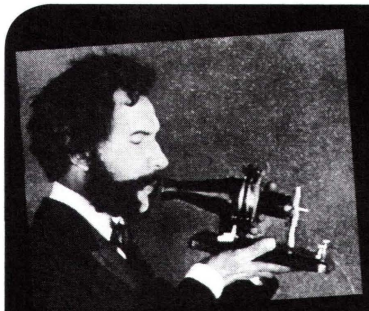
NEW BLUE BOX SHIRT



store.2600.com
$20

We've retired the "blue" blue box shirts and have gone back to our roots with the traditional white on black style. Not only is it more readable, but it washes better and will last forever (we still see people with the ones we made over ten years ago). It also has brand new headlines on the back relevant to the hacker world.





TELECOM INFORMER

by The Prophet

Hello, and greetings from the Central Office! As winter thaws into spring, I'm writing this column as I prepare for a trip to my first subcontinent - India! A topic for a future column, we're closing our call center in India and I'm heading there to wind it down. There is some incredible new technology that allows us to bring call centers back onshore with a lower total cost (and higher customer satisfaction) versus offshore. But before I jump on yet another plane, here's what I was up to over the winter. Another quarter has taken me to another two continents, this time Europe and Asia. In fact, I was back and forth between Europe and Asia a few times each day at one point. How? By crossing over the Bosphorus, a strait which bisects Istanbul and separates Europe and Asia. As this was my first visit to Turkey, a country of 75 million people bridging Europe and Asia, I was obviously interested in what I could learn about telecommunications there.

The weather in Istanbul was awful - sideways sleet - so I had the perfect opportunity to enjoy a "phone trip." I used to pay a lot of attention to payphones when I traveled, but I quickly gave up in Turkey. As in the U.S., most of the payphones that I saw were either vandalized or out of order (although you will find the occasional working one). At no point did I see anyone actually using them, either. It seems that this is an increasingly common condition. Payphones are still scrupulously maintained and meticulously cleaned in Japan, but in developed countries, it's increasingly rare to find working payphones. Developing countries still do a brisk business in public phones, but most of these aren't payphones. Instead, most public phones are in Internet cafes and calling shops. I saw an Internet café in Istanbul offering calling services to the public, but only one. Public phones just don't seem to be very popular in Istanbul - at least in the parts of the city I visited.

The lack of public calling services was somewhat surprising given the very high cost of mobile phones in Turkey. One of my first stops when getting off a plane in a new country is at a mobile phone kiosk or vending machine. It's so common to change SIM cards when you change countries in Europe and Asia that there is almost always a convenient place to buy a SIM card at the airport. Istanbul is no exception. I flew into Sabiha Gökçen airport, and there is a Vodafone kiosk just outside the baggage claim. I stopped by and asked how much a SIM card cost, and was quoted an outrageous sum - the equivalent of about $75 US! I was incredulous, and asked whether that was just for the SIM card or whether it included service. A month of service was, in fact, included, but it was nothing to write home about - an hour or so of local calls, and 500MB of data service. The Vodafone representative assured me that this was price competitive with other providers in the area, and when I asked why it was so expensive, he gave me a blank look and what I eventually dubbed the "Turkish shrug." This is because in Turkey, people often don't ask why, know why, or want to know why; instead, they just give you a blank stare and a disinterested shrug.

Obviously, paying $75 for a SIM card wasn't going to work for me (I was only staying for three days), but I was curious why it was so expensive. As it turns out, in Turkey, there is a high tax on mobile phones. These are registered with a local tax agency and the IMSI (a unique code that identifies your equipment) is registered with the Turkish tax authorities. All mobile phone providers are required to validate that your

phone's IMSI exists in the tax authority's database before allowing you to connect to the network using a local SIM card. Vodafone actually handles this process for you if you buy a SIM card at the airport (and the cost is included in the SIM card), but otherwise, you have to fill out forms in Turkish and take them to a government office in Istanbul along with your tax payment. Only then are you allowed access to the Turkish mobile network on a local SIM card. While it is possible to roam, roaming charges can be very expensive in Turkey. Using my Dutch SIM card, local calls cost over three euro per minute in Istanbul, and SMS messages were 85 euro cents each.

"Fine, I'll just use Wi-Fi," I thought. Unfortunately, when I arrived at the hotel, I discovered another reality of connectivity in Turkey: the Internet is censored. It's not censored to the degree of China, but it's cranked up a notch above Thailand. Many sites I routinely visit were blocked - things like Reddit, Gawker, and even technical publications (although 2600.com was not blocked). While I was told that Internet blocking isn't as extensive as it was during the Gezi Park protests (where Twitter, along with many other social media sites, was blocked), a lot of the blocks implemented during this time never went away.

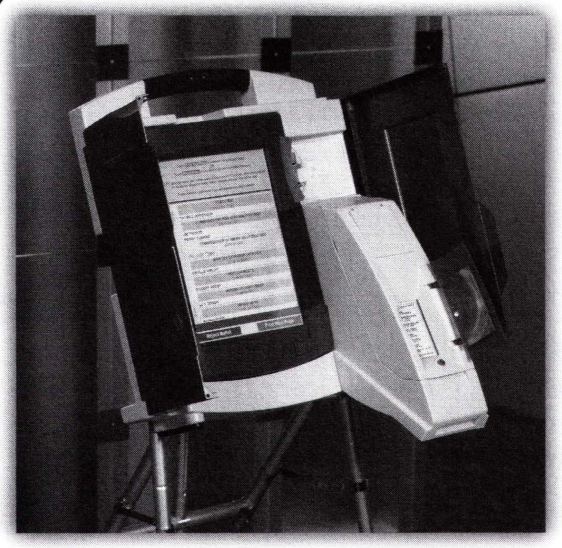
Normally I can get around these problems with a VPN, but the hotel Wi-Fi blocked these! So, in searching around for a solution, I discovered that it's possible to rent a mobile wireless hotspot in Istanbul. This is a business model that I haven't seen anywhere else in the world, but is necessary because of the massive headaches Turkish authorities impose on foreign mobile phone users. Cello Mobile and AllDayWifi offer this service, so I rented a hotspot from AllDayWifi. It cost about $7 per day, was delivered to my hotel, and came with unlimited data service. This ultimately solved both problems because my VPN then worked, and I was able to bring data service along with me (so I could use Google Maps, etc. on my Android phone).

When I rented the mobile hotspot, I was surprised to discover inconsistencies between what is blocked depending on which network is in use. The mobile hotspot I rented used the network of the smallest mobile phone provider in Turkey, Avea (the other two are Vodafone and Turkcell). At the hotel, the service was provided by Turk Telecom, the incumbent wireline provider. The differences appear to be because of how censorship is performed in Turkey. Rather than operating a firewall itself (like the Chinese government does), the Turkish government puts blocking responsibility onto Internet providers. This means that blocking can be a very blunt instrument if poorly implemented. Turk Telecom appears to block an entire website if any individual thing on it is required to be blocked, while Avea blocks just the individual page with content the government finds objectionable. So, although my VPN worked correctly, I found myself not needing to actually use it when connected to the Avea hotspot (apparently the stuff I read isn't anything the Turkish government wants to block).

I was interested to know what mobile phones Turkish people prefer, and what they do with them. The most popular mobile phone operating system in Turkey is Android, with an overwhelming preference for sleek, high-end, and aspirational models from Samsung. Phones in Turkey are sold unlocked, making a high-end mobile phone a substantial upfront investment for Turkish people and lessening the popularity of more expensive iOS phones. The app ecosystem is largely what you see in other European countries, but Turkish people love local music and entertainment apps. One guy showed me his phone with over a dozen of these installed. Given their shared language with other countries in the region, not all of these are Turkish-based. One music download app popular in Turkey is based in Azerbaijan for, I was told, "copyright reasons."

And with that, I'm hearing the final boarding call for my flight to Dubai, with onward service to Mumbai. Have an incredible spring and I'll write to you again this summer!

Brazil's Electronic Voting Booth



by Overall

Hello dear fellows, Brazil speaking here. Brazil is that big country south of Mexico with Carnaval, pretty women, Futebol (not soccer, please!!), and the third biggest democracy in the whole world - the only one I know with full 100 percent digital voting using an "electronic booth." Recently, we had our seventh free election in our recent history and everything went fine and good. Right? No, not right at all. I don't know what you guys heard - or not - about our elections, but let me give you just a little background information on how things are done here. I'll get to the hacking part really soon.

We have a presidential type of government with free and direct elections every four years. It's different from what happens in the United States. Here we vote directly for our candidate, meaning that each individual vote counts. So each one of 150 million votes is counted before we know the results of the elections. Sounds complicated, right? That's where our "Urna Eletronica" or electronic booth comes to the scene. As I'm a lazy man, from now on I'll call the Electronic Booth simply EB.

Brazil has 3038 electing zones in the entire national territory and each and every one of these receives two booths minimum, but most part of them have four or more booths. We have almost 142 million registered voters and a bonus: obligatory voting!

Yes, guys... in spite of being a democracy, we have a "forced" vote and military service.

Anyway, some 20 years ago, we used to wait a few days, if not weeks, to know the winning candidate, because each vote was manually counted, recounted, computed, checked, validated, and finally inserted into a vote database. I'm not kidding or exaggerating; that was our actual counting system.

Then some smart guy created and managed to sell to the government an electronic system that made everything easier and faster. It actually worked! And very well indeed, as now we can know an election's result in a matter of hours. This year, *all* votes were processed before midnight of Election Day. This was like magic, considering that all of the polling places were open until 6 pm.

The Brazilian government launched a big campaign before the elections concerning the safety of our great and famous EB. It is safe, they said. It is secure, it is good, and it is "unhackable...."

Yeah, right.

How Does It Work?

Our booth uses a system called Direct Recording Electronic (DRE) made by Diebold that has already been judged as unsafe in the U.S. in 2007, Holland in 2008, Paraguay in 2008 (fer-crying-out-loud! Paraguay?), and declared unconstitutional by Germany in 2009. It is based on Windows CE... [deep breath]

You go to the election zone you have been assigned to, handling your elector document, previously registered and authorized. "They" know who you are. (This year, you could bring your personal ID or driving license as well - everything is already integrated.)

Anyway, you go to the zone and show your document. The person gets it, finds you in the list, tells you to sign some ridiculously small paper that you *must keep*, or you cannot ask for a passport or buy big things, such as a house

or car. After that, some other guy *inserts* your elector number into a small numeric keyboard attached to the EB to "release it for vote." OK... I guess some of you smart guys already found a "flaw" in the process, but let's keep it going. There is more.

You go to the EB, chose your candidate by typing his/her number (sometimes "its" number may be valid... there are some really weird, well, things, here in Brazil). After you press the last candidate number and confirm the vote, you go home happily, knowing that you contributed to our good democracy.

What Happens Next is the Fun Part

Until now, the electronic booth was really hard to hack, as there is really good security surrounding all of the processes of building, configuring, and deploying the booth. Even if some guy or group manages to hack some of the EBs, it is not possible to make a big difference and change the election itself. Except that the entire process is run by a contractor, paid by the Brazilian Elections Justice system (separate from "regular" justice), who is paid by the government itself. In theory, some person with ill intentions working for this company might be able to add some algorithm that changes some votes, giving them to some other candidate before the software is inserted into the booths.

When the election is over, each EB is taken to a regional computing center, previously prepared to transfer the votes to a central processor by Internet. Here is the second hard-but-possible flaw: *All* booths carry the very same cryptographic keys. Yes, all of almost half a million EBs are using the same key pair. Those who have access to this key will be able to intercept, change, repack, and send the "fixed" package forward to be accounted.

Moving on.

After the transfer, vote counting is broadcast live on open TV and it is quite a show! But this is another point of weakness. The votes are transferred to a "black room," controlled by god-knows-who, handling the data while it arrives. A good DBA or a bad-boy hacker would be able, at this point, to change the data being inserted, as it has already been previously checked, so nobody checks it in the database, meaning that you can add some insert procedure to play with the numbers. Just to feed your creativity, this guy/group can change one vote in 50 from one candidate to some other during data insertion. No one will ever know.

In 2012, there was actually a successful hacking in Rio de Janeiro, where this hacker known only by his nick "Rangel" (along with his "friends") was able to intercept and change votes during the transmission process, simply by accessing the unsafe and easily hackable Justice Department regional Intranet in Rio. They actually made their candidate (or client) a winner.

Another very well known and documented case was when the Justice Ministry hired Brasilia University to run some security tests on the EB. The guys actually achieved a security break in the system, modifying the source code. The government said it was not a big concern, as the tests were made in a controlled environment and in "real life" it would not be possible.... Yeah, right.

There are some other points of concern:

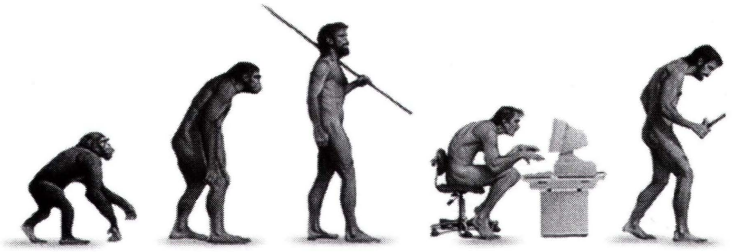
- You can actually know who voted for whom, simply by ordering the votes by date/time and checking who "logged in" at that time. Remember the guy who inserts your ID into the numeric keyboard? They know who you are.
- EB stores your vote directly into a smart card, and it is not possible to know if the stored number is the same as what you typed.
- As you cannot check what was inserted into memory, 50+ countries have already rejected the EB, some of which I have already mentioned above.
- In 2012, there were possible fraud cases in 94 cities, 30 in São Paulo State alone.
- Diebold (the EB manufacturer) was fined $50 million by the U.S. Department of Justice due to corruption accusations in other countries.

Well, all of this is fun, but that's not what concerns me the most. Brazil does *not* have a strong democracy just yet. We had a military government until "yesterday," as our first post war democratic election was in 1989 and our current government is mainly composed of ex-terrorists and rebel warriors who fought against the military government. It means that there are still some open wounds that must be amended.

As described, any dictatorial government may be able to know who you voted for. From this to "the wall" or to "the camps" is a quick step.

Bottom line: anyone who has access to the central election database controls its results.

EVOLUTION OF A HACK



by Uriah C.

Some years ago, I was learning the basics of an SMTP server. I learned how the protocol allowed me to connect and I could interact with it. Well, the first thing I did was to send myself an email. I just happened to send that email saying that it was from santa@northpoll.com. My first spoofed email was created, and sent.

Over time, I wanted to be able to do this more often. I installed sendmail on my Linux box, and started to telnet to my local server. It worked great. I knew how to make the email say it was from anyone. I put my outgoing server settings on my mail clients to my local server so I didn't have to login to send mail.

Then I had an idea for a prank! What would someone do if they got 20 emails from themselves? So I fired up my text editor and wrote down my first mailFlood.py program. It was run in the console and asked for input and did the job. The pseudocode went like this:

```
mailFlood 1

Write "How many emails:"
num = read input

Write "what is the Address:"
addy = read input

subj = "PWND"
body = "Stop emailing yourself!"

date = system.get_date

msg = "From: %s\nTo: %s\nSubject
➥: %s\nDate: %s\n%s\n
    %(addy, addy, subj, date,
➥ body)

n = 0
while n < num:
    connect to smtp (localhost,
➥ 25)
    sendmail (addy, addy, msg)
    disconnect from smtp
    n = n +1
```

A very simple way of doing it. Once I coded it up in Python, I got it working with no problem. Later, I thought I would like to change the message a bit. I want to have a personal message, other "From" email address, and so on. I revised my code a bit. The pseudocode was like this:

```
mailFlood 2

define numb():
    write "how many emails:"
    numb = read input
    return numb

define fromAddy():
    write "From address:"
    fromAddy = read input
    return fromAddy

define toAddy():
    write "To address:"
    toAddy = read input
    return toAddy

define subj():
    write "Subject:"
    subj = read input
    return subj

define msgBody():
    write "type your Msg:"
    msgBody = read input
    return msgBody

define main():
    num = numb()
    fromAdd = fromAddy()
    toAdd = toAddy()
    sub = subj()
    msgBod = msgBody()
    date = system.Get_date

    msg = "From: %s\nTo: %s\nSub
➥ject: %s\nDate: %s\n%s\n
        %(fromAdd, toAdd, sub,
➥ date, msgBod)

    n = 0
```

```
    while n < num:
        connect to smtp (localhost, 25)
        sendmail (fromAdd, toAdd, msg)
        disconnect from smtp
        n = n +1

main()
```

It was a bit longer now, and I could have saved code by not defining all the inputs on their own. I did have a reason for doing this. I was thinking about errors, and what if someone put a string when it asked for the number of emails. I could later put in a check for an integer, without changing the main function.

Python had upgraded a lot since I last translated that pseudocode. Also, Python 3 is not backward compatible with 2.x. I decided that it was time to revamp the code one more time. Since my program uses interactive prompts instead of command line arguments, I decided that I should give it a GUI.

The nice thing about the last pseudocode is that it is read to be converted to an event driven GUI. I had not planned it that way, but it was just there when I dug it up to recode. So here is a pseudocode of the GUI program:

```
mailFlood 3

define window():
    window = GUI()
    numblbl = Label("Number of Emails")
    numtxt = Entry()

    fromlbl = Label("From:")
    fromtxt = Entry()

    tolbl = Label("To:")
    totxt = Entry()

    sublbl = Label("Subject:")
    subtxt = Entry()

    bodylbl = Label("Body:")
    bodytxt = Entry()

    sendbutton =  Button("Send", cmd=onClicked)

    window.mainloop()

define onClicked():
    num = int(numtxt.get())
    fromadd = string(fromtxt.get())
    toadd = string(totxt.get())
    subj = string(subtxt.get())
    body = string(bodytxt.get())
    date = system.get_date

    msg = "From: %s\nTo: %s\nSubject: %s\nDate: %s\n%s\n
        %(fromadd, toadd, subj, date, body)

    n = 0

        while n < num:
            connect to smtp (localhost, 25)
            sendmail (fromadd, toadd, msg)
            disconnect from smtp
            n = n +1

window()
```

Now that I had the basic idea of what I wanted, I could then refine it a bit. The basic program logic is the same: put user input into a format the SMTP server can use, start a counter, connect and send message, add one to the counter, repeat. The bulk of the code is formatting the GUI, while I could almost cut and paste the main logic from version to version. I ended up coding it in Python 3, and using Tkinter for the GUI. I spent a good amount of code framing up the window so it would have a nice UI to look at.

When I hacked this joke up back in 2010, I had no idea that I would still be working on it in 2014. One can write a joke program like this one, a script that pulls log files, or a quick and simple server that lets you connect to a computer remotely for some reason. You may need to update the code, add a feature, or fix a bug. The point is, you don't know how a simple thing will grow more complex, and actually be a project in development.

I started with telnet, went to a CLI based program, then to a CLI based program with more options, and finally to a full GUI program. It is obvious that there was some evolution there. That evolution was not just the code. It was an evolution in me. As I evolved as a hacker, my coding evolved. Let us hack away. Solve problems, and learn about systems. See what breaks, and then try to fix it. And never forget the reason we hack... it's *fun!!!*

And, in case you wanted to see it, here is the actual Python 3 code I have:

mailFlood.py

```
#start of code
#import smtp functions, date functions, and Tkinter
from smtplib import SMTP
import datetime
from tkinter import *


class mf22:

    def __init__(self):

        #make the GUI
        self.window1 = Tk()

        #frames to section the GUI and improve layout
        self.frameTop = Frame((self.window1))
        self.frameTop.pack(side=TOP)
        self.frameBot = Frame((self.window1))
        self.frameBot.pack(side=BOTTOM)
        self.frame1 = Frame((self.frameTop))
        self.frame1.pack(side=LEFT)
        self.frame2 = Frame((self.frameTop))
        self.frame2.pack(side=RIGHT)
        self.frame3 = Frame((self.frameBot))
        self.frame3.pack()

        #Put in the labels and texfields and the button
        self.numLbl = Label((self.frame1), text="Number of Emails:")
        self.numLbl.pack()
        self.numTxt = Entry((self.frame2), text="")
        self.numTxt.pack()

        self.fromLbl = Label((self.frame1), text="From:")
        self.fromLbl.pack()
        self.fromTxt = Entry((self.frame2), text="")
        self.fromTxt.pack()

        self.toLbl = Label((self.frame1), text="To:")
        self.toLbl.pack()
        self.toTxt = Entry((self.frame2), text="")
        self.toTxt.pack()
```

```
        self.subLbl = Label((self.frame3), text="Subject:")
        self.subLbl.pack()
        self.subTxt = Entry((self.frame3), text="")
        self.subTxt.pack()

        self.msgLbl = Label((self.frame3), text="Message:")
        self.msgLbl.pack()
        self.msgTxt = Entry((self.frame3), text="")
        self.msgTxt.pack()

        self.sendButton = Button((self.frame3), text="Send",
            command=(self.onClicked)) #button
activates the event onClicked
        self.sendButton.pack()

        self.doneLabel = Label(self.frame3,
            text="All fields will clear when done")
        self.doneLabel.pack()

        #start the main loop
        self.window1.mainloop()

    def onClicked(self):

        #pull data from the entries
        self.num = int(self.numTxt.get())
        self.fromAdd = str(self.fromTxt.get())
        self.toAdd = str(self.toTxt.get())
        self.subj = str(self.subTxt.get())
        self.body = str(self.msgTxt.get())

        #get the time and date
        self.date = datetime.datetime.now().strftime("%d/%m/%Y %H:%M")

        #format the message
        self.msg = "From: %s\nTo: %s\nSubject: %s\nDate: %s\n\n%s" \
        % ((self.fromAdd), (self.toAdd), (self.subj), (self.date),
➥ (self.body))

        #start the counter and be ready to connect to the mail server
        smtp = SMTP()
        n = 0
        l = self.num

        while n < l:
                smtp.connect('127.0.0.1', 25)
                smtp.helo('localhost')
                smtp.sendmail((self.fromAdd), (self.toAdd), (self.msg))
                smtp.quit()
                n = n + 1

        #clear all fields once done
        self.numTxt.delete(0, END)
        self.fromTxt.delete(0, END)
        self.toTxt.delete(0, END)
        self.subTxt.delete(0, END)
        self.msgTxt.delete(0, END)

#run the class
if __name__ == "__main__":
    main = mf22()
#end of code
```

Bleeper - Downloading Full-length Preview MP3s from bleep.com

by Derek Kurth
dkurth@gmail.com

Bleep.com is an online store for indie music, especially electronic music. On every album's page on the site, you can listen to the entire album in 30-second increments. To accomplish this, Bleep is actually sending the entire song's MP3 to your browser, then using JavaScript to cut you off after 30 seconds. By monitoring HTTP requests in your browser, you can see how the entire MP3 is being downloaded to your local cache.

Note that these MP3s are lower quality mono recordings. Bleep sells much better lossless (or high bit-rate MP3) versions on the site.

Every album has a URL that looks like this:

```
https://bleep.com/release/55391-
➥jon-hopkins-asleep-versions
```

Note the numeric ID, 55391. We'll use that in a minute.

Load that page in Chrome, then open the Developer tools (from the Chrome menu > More tools > Developer tools). Click the Network tab in the dev tools, then click the play button next to the first song. You will see a few network requests, but the important one has this URL:

```
https://bleep.com/player/resolve
➥/55391-1-1
```

Notice how this URL includes that same album ID, 55391. The response to that request is just a URL that looks like this:

```
http://preview.bleep.com/f1fd2b6
➥a-de5f-3d28-b061-87fe8a12f892-0
➥1-001.mp3
```

That is the URL for the entire preview mp3! It looks like Bleep assigns a UUID (in this case, "f1fd2b6a-de5f-3d28-b061-87fe8a12f892") to every album, and the number at the end corresponds to the track number. So, if you want the mp3 for the album's second track, just change the "001" at the end to "002", and so on for the other tracks. (I could not figure out what the "01" between that UUID and the track number is for, though.)

At this point, if you start downloading previews, you'll quickly realize that 1) it's tedious, and 2) the filenames are those inscrutable UUIDs. So, we'll write a Python script for pulling down an entire album, naming the files using the correct artist, album, and track names. (N.B., all the code in this article was written for Python version 3.4.)

We'll create two classes: BleepAlbum and BleepTrack. A BleepAlbum has an id, artist, title, and a list of tracks. Each track in the list will be a BleepTrack, which has a URL, an index (the track number within the album), and a download method.

Also, since we don't want to type in the artist name, album title, and track titles, we'll scrape them from the album page's HTML. When we're done, we'll be able to download the preview mp3s for an entire album with just three lines of Python, like this:

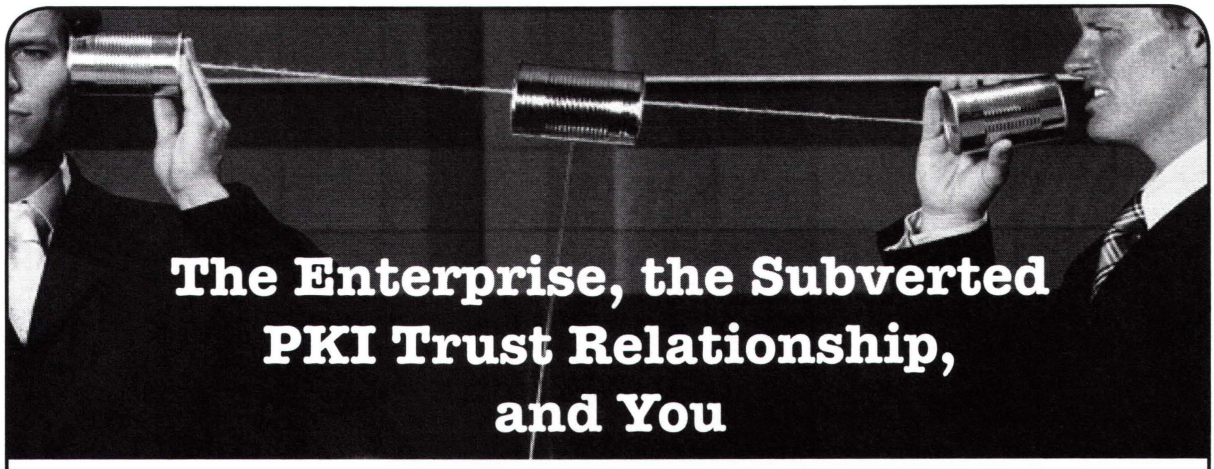
```
album = BleepAlbum(55391)
for track in album.tracks:
    track.download()
```

We are going to use two Python libraries that you might not have: Beautiful Soup and requests. You might need to run "pip install beautifulsoup4" and "pip install requests" before these imports will work.

Here is the full code for pulling down an album. I left out the code to catch exceptions and to accept an album ID from the command line - those are "left as an exercise to the reader," as they say. But if you save this as bleeper.py, set the album_id variable (near the bottom) to whatever album you want to download, and run "python bleeper.py." It should work. If Bleep changes the structure of the album page, the parsing for the titles will need to change, also.

```
### Code begins here

import requests
import re
import os
from bs4 import BeautifulSoup

class BleepTrack:
  def __init__(self, album, title,
➥ index):
    self.album = album
    self.title = title
    i = str(index)
    self.padded_index = "0" *
➥ (3-len(i)) + i # turn "5" into
➥ "005"

  @property
  def url(self):
    return self.album.url_prefix +
➥ self.padded_index + '.mp3'

  def download(self, dest_dir="."
➥):
```

```
    r = requests.get(self.url,
➥ stream=True)
    dest_file = os.path.join(dest_dir, self.padded_index + " - " +
➥ self.title) + ".mp3"
    with open(dest_file, 'wb') as f:
      for chunk in r.iter_content(chunk_size=1024):
        if chunk: # filter out keep-alive new chunks
          f.write(chunk)
          f.flush()

class BleepAlbum:

  def __init__(self, release_id):
    self.release_id = release_id
    album_url = "https://bleep.com/release/{}".format(self.release_id)
    r = requests.get(album_url)
    self.soup = BeautifulSoup(r.text)

  @property
  def artist(self):
    details = self.soup.find("div", "product-details")
    artist = details.find(itemprop="name")
    return artist.text

  @property
  def title(self):
    details = self.soup.find("div", "product-details")
    album_wrapper = details.find("dt", "release-title")
    title_elt = album_wrapper.find("a")
    return title_elt.text

  @property
  def tracks(self):
    tracks = []
    index = 1
    for span in self.soup.find_all("span", "track-name"):
      title = span.find("a").attrs['title']
      title = re.sub("^Play '(.*)'", r"\1", title)
      track = BleepTrack(self, title, index)
      tracks.append(track)
      index += 1
    return tracks

  @property
  def url_prefix(self):
    try:
      return self._url_prefix
    except:
      pass
    get_preview_url = 'https://bleep.com/player/resolve/{}-1-1'
➥.format(self.release_id)
    r = requests.get(get_preview_url)
    preview_url = r.text
    match = re.search("(^http://.*?-01-)\d+.mp3", preview_url)
    self._url_prefix = match.group(1)
    return self._url_prefix

if __name__ == "__main__":
  album = BleepAlbum(55082)

  # This creates a directory like "Artist name/Album title", and saves
➥ the tracks there.

  save_path = os.path.join(album.artist, album.title)
  try:
    os.makedirs(save_path)
  except FileExistsError:
    pass
  for track in album.tracks:
    print(track.title)
    track.download(save_path)
```



The Enterprise, the Subverted PKI Trust Relationship, and You

by Mike
mike@tofet.net

A Sinister Plot

I work for a large, buttoned-down, conservative organization that frowns upon unreadable network traffic. This organization had decided it wanted to be able to decrypt all HTTPS web traffic so it would be inspectable. The main reason for this was to prevent data leaks of sensitive, proprietary, or privacy-act related information. Part of our job contract and our network user agreement at this organization is the explicit acceptance of constant monitoring. We know this from the start, so we shouldn't expect privacy.

But when we go to an SSL-protected website and see our little friend the padlock, we do expect a reasonable level of privacy. For me, the only information I generate from my work computer that I want to protect are the passwords to my banking, credit cards, utility accounts, etc. I don't spend much time doing this sort of thing at work, but the fact remains if you want to contact someone during business hours about your account, you are likely doing it during *your* business hours as well.

Being in a bit of a leadership role at this organization, I was privy to the initial plan put forth to open up the SSL traffic to inspection. Without going into details, I knew the initial plan was doomed to failure. I dutifully pointed out the flaws, but was reassigned before I got insight into the final plan. To keep the growing hacker outrage I can already hear in check, I had known about the reassignment before my review of the plan. Nevertheless, I know with certainty that the organization will pursue the goal of inspectable SSL/TLS traffic until they get what they are looking for.

So How *Do* You Do It?

The only really reliable way of inspecting SSL/TLS traffic is to conduct man-in-the-middle (MITM) observations. In short, this means all traffic leaving the client must go through a middleman before continuing on to the distant server. These middlemen are called proxy servers, or gateways in network parlance. It is trivial to set up a man-in-the-middle proxy server capable of interjecting itself into the SSL flow. As an example, check out pymiproxy.[1]

Right out of the box, this very compact Python-based proxy server will decode all SSL traffic and give you full access to the traffic for logging, exploration, password sniffing, et cetera. pymiproxy is just one such tool that can do this with ease. There are also very popular tools such as sslsniff and sslstrip available; the choice is a matter of preference and environment.

But there is a problem. If you set up your MITM proxy server, point your browser to it, and then surf to an SSL encrypted page (such as google.com), you will almost certainly get a big glaring error page. This page will say something similar to: "The certificate for this web page cannot be trusted. We recommend you leave town immediately. Or, you can do a bunch of mouse clicking to tell us you know what you are talking about."

This is because the MITM proxy server has violated trusted Public Key Infrastructure (PKI). On the way to Google's servers, your request for a secure web connection was intercepted and repackaged as though the proxy server itself was making the connection request. Google's servers dutifully sent a very expensive, special, cryptographically-signed identity certificate to the proxy server. The proxy server used this certificate to establish a secure connection with Google. Then, the proxy server generated a free, less-special, but still crypto-

graphically-signed identity certificate to your browser. Well, your browser can smell a rat and lets you know before establishing a connection.

Your browser can detect the difference because Google's certificate was protected by GeoTrust's Global Certificate Authority (CA) cryptographic signature. Your browser was taught from birth that this signature is digital gold and anything signed by it must glitter. The MITM proxy's certificate was signed by an automatically-generated CA that your browser doesn't know. So it gets mad, warns you, and you run away screaming.

This difference between known Certificate Authorities and rather wonky ones is the root of trusted Public Key Infrastructure. Your browser has been taught that a select few certificates - though more than you probably realize - should be trusted implicitly; anything else is likely evil.

Whew!
That Should Protect Me! Right?

No. Your browser can be taught that any given certificate should be trusted as much as GeoTrusts' are. The functionality to add, delete, modify, and otherwise manage trusted certificates is built into every modern browser and operating system. Simply google "manage certificates" followed by your browser name and you'll get a nice list of tutorials.

In this case, if the CA used by the MITM proxy server is trusted by your browser, there will be almost no visual warning that anything is amiss. You will connect to Google, complete with a nice padlock, and conduct your searches in blissful ignorance. In the meantime, everything you send is being decrypted and logged before being sent on to Google's servers. There are some differences in the presentation of the security indicators, but there will be no large, glaring warning such as the one you originally saw. You, as a reader of this magazine, may catch the difference, but I can guarantee you your coworkers won't.

You can see this difference in action by setting up a known CA, having your browser trust it, and then making the MITM proxy use it.[2]

Got It.
Don't Trust Strange Certificates.

This is where the enterprise comes in. In this sense, enterprise is short for: "We, the corporate entity, control every aspect of your computing environment to include hardware, software, configuration, and connections." Most large business entities have standardized computer deployments; everyone uses the exact same hardware and exact same initial software setup, typically cloned from a Norton GHOST image. This is, in fact, a security best practice because it greatly simplifies the patching and update process.

If that standard configuration comes with a pre-loaded, company-controlled Certificate Authority and that CA is used to sign the identity certificates received by your browser, you may not ever realize your SSL-encrypted communications are being watched. The enterprise has subverted the PKI trust relationship by trusting a CA for you.

That Sucks. How Can I Know?

The easiest way to know is to examine the CA path of your most commonly used websites from a trusted computer and network location. Make note of the name of the root CA used and then compare this to the root CA when you use the website from a less-trusted location. If the root CA matches, then there is a very high probability that the end-to-end SSL connection to your website is secure. In short, you have to become more vigilant in your use of the technology and not simply trust the security indicators.

But this is also easy to subvert. When you make your own CA signature, you can literally name it anything you want. You could name it "GeoTrust Global CA" and install it in your browser. Then anything signed by this CA will have a root CA of "GeoTrust Global CA."[3] I tested this in Firefox on Ubuntu and it worked like a charm. The trusty lock icon turns from a full color picture to a mere gray shadow of security, but there are no large errors. Firefox didn't even complain when I imported the certificate file. Instead, it filed the new cert under the actual GeoTrust Inc. group in the certificate manager!

You could make one of these up for all of the major CA companies out there and put them in the browser's certificate store. It would be trivial to program your MITM server to send the appropriate fake root CA depending on what CA was sent by the distant server. If all you do is check the name of the CA, you may once again not realize you are being had.

The next level up from checking the names individually is to visit a few secure websites and note the root CA used. If it is the same root

CA for each website, odds are there is a man-in-the-middle proxy involved. For instance, Google currently uses GeoTrust Global CA but Yahoo uses "VeriSign Class 3 Secure Server CA - G3" as their root CA right now. But, if I surf to Yahoo through my MITM proxy, it says GeoTrust Global CA. If you visit four or five sites and they all have the same root CA, your SSL traffic is being inspected. This method isn't foolproof, of course, since the MITM proxy could serve up an appropriately named CA for each site as already discussed.

The only surefire way to detect a MITM proxy is to examine one of the mathematically calculated hex-encoded field values that all certificates have. There are several to choose from, but the one that would absolutely ensure you are using the correct certificate is the public key. If the public key matches every time, then it is the correct certificate. You wouldn't need to memorize and check the whole public key - perhaps just the first eight bytes and the last eight bytes would be enough to ensure the certificate was right. If they do match, I would say you are in very good shape, crypto wise.

Although it might seem daunting to make this check, you don't need to check every site you use. If you do the public key check on two popular websites that default to SSL - say google.com and yahoo.com - and everything checks out, odds are very high there is no MITM SSL-inspection proxy server at work.

As an example, when I surf to google.com without a MITM proxy in the way, the first eight bytes of the GeoTrust Global CA certificate are: DA CC 18 63 30 FD F4 17. If I surf to Google through my MITM proxy with the fake GeoTrust Global CA certificate discussed above, the first eight bytes of the public key are: F0 0A 93 56 DE DB 4F 49. You can see I didn't even need eight bytes to make this determination. Just one byte is enough to tell me they are different.

The only real downside to this method is when sites change their certificates. Certificates expire, of course, or a site may change root CA providers to get a better deal. So you need to spend a little time making sure you stay up to date on the current public keys of your test sites. But really, that isn't much of a price to pay for a little sense of security.

Moral of the Story

A MITM SSL inspection proxy is just one of the ways a corporate enterprise can reduce the security level of otherwise secure computer use. Since they control your entire desktop, they could easily put in key loggers or other software that gives them insight into all of your activity.

Since you are working on their networks, under their constraints, and they are paying you, there isn't much you can do about it. But, you do have the right to know your secure information isn't so secure. Since they aren't likely to tell you, you need to come up with techniques like the ones discussed above to detect their intrusion.

Update: You're not safe away from the enterprise either. A few months after I wrote this article, a story broke about Lenovo and Superfish. It turns out certain models of Lenovo laptops had a piece of adware/malware installed at the factory called "Superfish." Superfish did many things, but the most insidious was installing a pre-trusted root CA. This root CA could then be used to spoof secure connections in exactly the way I discussed. Bottom line: you need to stay alert no matter what computer you are using.

Notes

[1] pymiproxy can be downloaded from GitHub: `https://github.com/allfro/pymiproxy.git`

[2] I used the instructions at: `http://www.davidpashley.com/articles/becoming-a-x-509-certificate-authority/`

to set up my CA. I had to make the following changes to the openssl.cnf to get it to work reliably:

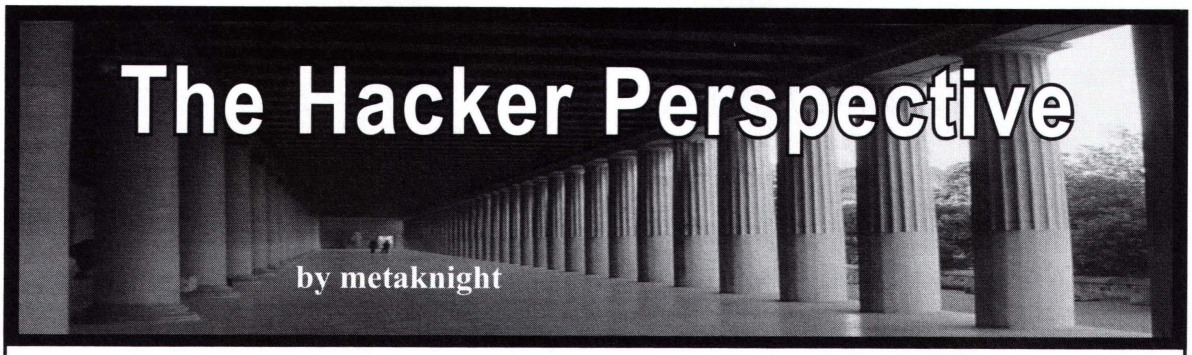
```
[ policy_match ]
countryName = optional
stateOrProvinceName = optional
organizationName = optional
organizationalUnitName = optional
commonName = supplied
emailAddress = optional
```

Once the pem and key files have been made, you need to concatenate the files into a single pem file. You can then start pymiproxy with: `python proxy.py yourCAfile.pem`

Make sure you import the pem file into your browser and configure the trust settings correctly.

[3] You can set your certificate to any name you like with the "organizationName" and "commonName" fields in the openssl.cnf file.

The Hacker Perspective



by metaknight

We waited for the clock to signal the end of detention. The second hand moved as though it weighed ten stone three. This being the early nineties, security cameras and guards were not yet a paranoid fixation of high school administration. So when our period of enforced life wasting expired, we dispensed ourselves from the tomb with alacrity. Out of the bunch, my friend Richie and I were on a mission. Our destination: the library. We were free to roam the halls of the otherwise empty school, dodging only a few administrators. Earlier that day, Rich gave me a 3½ inch disk covered in black sharpie and silver paint marker to hold. We were going to install its viral contents into the library computers. At this time, my only experience with computers was back in second grade (circa 1985) in a room filled with tan behemoths and screens covered in green text of yesterday's lore (what kind of computers?).

Anticlimactically, we snuck into the darkened library - a place I'd only been a few times in four years - and turned on the two computers (couldn't tell you what type - still off my radar) and Rich went about injecting whatever virus was on the disk. I later found out that it rendered the machines permanently unusable. Back then I believed such things. I had the disk in my possession for a few years after that, never used it (didn't have a computer until 1999), and ended up throwing it away.

Rich was also responsible for exposing me to two other interesting things, the first being the infamous "2600 box." In the cafeteria was a payphone. He would put the little speaker of the box to the microphone of the receiver, hit the 9 button on the box if I'm not mistaken (superimposed memory?), giving us tons of credits to call someone he knew out in California.

He also taught me that on any payphone you could make it call itself, ringing endlessly until someone picked it up. I forget the three numbers used - something like 259 or whatever. In any case, one would get the dial tone, dial the three numbers, hang up the phone, pick it up for half a second, and then leave it hung up. The phone would then ring forever. This was endless fun in public places when you're a group of teens hell-bent on causing trouble. "Hello... helloooo!?" Watching people get annoyed at no answer was hysterical. Redial and repeat.

I am metaknight and I am a hacker. Yes, I am named after the enemy character in *Kirby*. My sword is usually a USB drive and my armor is anonymity. Metadata is my energy source. I explore networks like one would explore a Castlevania map, though by nature I am an audio engineer. If you met me, you would think this description juxtapose. (Well, perhaps not the readers of this zine.) So many people not in the know either associate hackers with criminals, as they are routinely trained to do by mass media, or with skinny, glass wearing geeks. Well, I used to think that hackers were just nerdy types. Only after having my ignorance of the proper definition removed, as is often the effect of education, did I come to have a broader understanding of what hacker means.

The above mentioned mischief with Rich does nothing to illustrate what a hacker is per se, but they were important seeds that were planted in my head that would significantly contribute to who I am today and how I've come to handle situations of varying degree with a hacker's mindset. I was part of a different group of kids in school. I did not spend very much time with Rich before we drifted apart, and since then I have never seen him again.

I come from a background with no money, divorced parents with a family history of alcoholism, depression, and suicides. I suffer from the on/off switch that suicidal thinking is, a switch that randomly does what it wants, regardless of what mood I'm in. After moving around from place to place until I was 13, I went to live at my rich uncle's house - a poor kid living in a rich neighborhood. Around the time of the library virus, I was transitioning from wigger to death metal kid - wearing Cannibal Corpse t-shirts with Karl Kani jeans - it just doesn't look right. Constantly being made fun of builds a strong will to avoid people and opens up one's self to his own creativity, along with a combination of having no money and having a stoner father that you have

no relationship with - even enough to get a ride to any friend's house miles away.

In my wiggerdom, I was pseudo-mixing nineties hip-hop records, making "mix tapes" with a setup I hacked together. Actually, it's not really at all an impressive hack, but it started me on my trek to bigger things. Anyway, I had to simulate the effect of two records going back and forth, something I figured out by listening to the way Funk Master Flex mixed on the radio, not from someone telling me that that's how it's done. I had one of those turntable-cassette deck-8-track combo units. So I would dub the record onto a cassette tape, run it, and rewind it in the player while using the *modeselektor* (wink) to switch back and forth between the record and the tape white mixing the record, simulating the sound of two records. With practice, I was able to nail the tempo and not ruin the mix. I used loose-leaf paper as a record slip. I lined out the mix to the line-in on my favorite Panasonic boom box and recorded mix tapes for people at school, which were admittedly terrible. Eventually I got a Gemini PMX-2500 4-channel mixer, a Sony SL-D2 direct drive turntable and a Technics 1200 from a friend - just in time to make a musical transition that led to my disinterest in rap.

I started to learn bass and guitar. Within six months, I learned how to play both well enough from playing along with my new favorite bands that I was starting to write my own material. I needed a way to record it, though. I wasn't able to form a band because I wasn't good enough to play with my amazing friends. I did not have a consistent ride to get anywhere anyway, and I couldn't afford to get bigger, better equipment in order to play live. Did I mention that I was playing death metal? No one wants to do that. It was time to hack together something.

I did not know what a 4-track was yet, but I knew I could make use of the mixer and the tape decks somehow. I connected the guitar to the distortion pedal, and its ¼ inch cable to channel 1 of the mixer and recorded the first guitar, panned left into the boom box. Then I took that tape, put it into the tape player on the turntable unit, ran it through channel 1 of the mixer while running a second guitar through channel 2, panned right into the boom box. Then I took that tape, switched it with the other again, and ran it through the mixer while recording bass. Now I had a full song with two guitars and a bass. But I did not have a drum set yet.

In my uncle's living room (the one no one actually sits in) was a Yamaha Clavinova electronic piano that had other instruments including drums in it. I took my setup into that room and ran a guitar cable from the piano's headphone jack into the mixer and dubbed drums into the mix. Bam! Now it sounded like an actual song. I took it all back to my room and overdubbed some terrible growls through a cheap ten dollar Realistic microphone from Radio Shack. I repeated this songwriting/recording process until I had a six song demo - recorded on my infinite track setup. I was ecstatic! I had never spoken the words "4-track," "recording studio," or "guitar lessons" in my life before this time. I instinctively knew to measure the j-card of a factory produced cassette, cut paper to size, and draw a band logo on it. Then I glued pictures to it, and ran off copies in the school's library copy machine, then distributed some of the finished demos to my friends. I was asked what band it was and when I said that I had done it all, they did not believe me. Some thought It was terrible and to hear it in comparison with what I can make today, it is embarrassing. But that's not the point. I consider this story a prime representation of the hacker's mindset. "But there was no computer involved!" the uninitiated would decree. Ah, but there was - the one we're all born with. For some it's turned on, for others it's not.

In writing this article, I thought I would use computer-related examples as illustrations. But when I rewound my life reel to the beginning of my hacking, there weren't yet any computers, and that seemed nostalgic and less pretentious as subject matter. I became a hacker out of a necessity to accomplish tasks with limited resources. I had to make things work that were intended for other purposes. Hackers aren't confined to computers.

Still high school era, I was suspended for a week out of school when I was accused of stealing another student's car with a group of friends. We all took turns driving it, neutral dropping it, etc. Actually, the reader will laugh. It was the aforementioned Rich's Cougar, long after we had drifted apart. I did not drive the car off of school grounds, so I did not steal it. The dean was particularly nasty, using colorful language forbidden towards students when addressing their misdeeds. Case in point: I needed to highlight both this and my argument of innocence to absolve myself from punishment by both the school and my apathetic, short-tempered father. I decided to rig up a way to record a phone call with the dean.

While on suspension and home alone, I experimented with ways to record phone calls. Using the Realistic microphone made too much feedback in the speakers and I needed more than two hands to accomplish being able to hear and record simultaneously. The phone was an old tan thing with push buttons and had a receiver attached

with the usual slinky wire (real technical descriptions, I know) that sat on two buttons each in their own U-shaped holder. I rigged up the receiver in a way that it was upside down on the base and so I would only have to tilt it to get the dial tone. I took a pair of over the ear Pioneer headphones, disconnected one side from the headband, and rubber banded it to the receiver, screwed on the ¼ inch adapter to its plug, jacked in to the mixer, hit record on my boom box, and called the school.

(For those who don't know, microphones and headphones are essentially the same thing, formatted for different applications; a small speaker in a headphone will act as a microphone when plugged into a mic input, but you'd blow a microphone trying to pump audio through it.)

I pretended to be someone else when I called the school in order to get the dean on the phone, which probably wasn't necessary, but still fun anyway. The dean answered the phone. I had to bait him past his apprehension of my calling him in order to argue something that, in his eyes, I was clearly guilty of doing. In defining my non-participation in the initial theft of the car, his temper was aroused and out came the venomous language, all perfectly captured on a 60 minute TDK cassette. This worked wonders for getting him reprimanded and suspended himself, but did nothing to get me out of trouble. As a result, I ended up missing out performing in the only high school play I ever would have acted in (*Guys and Dolls*), though my name remained on the printed rosters, and since my character ironically did a phone call in the play, something that would have made me invisible to the crowd anyway, I still received compliments on my stellar performance by people unaware of my absence. Kudos to Vas500 for covering that role. Eventually, I went on to record some hilarious prank call tapes by calling the help wanted.

I'm purposely sticking with high school era stories for a reason. The younger readers and newcomers to hacking will have more enthusiasm and open-mindedness than us old seasoned pros. At the same time, it's something I wanted to touch on in order to show that hackers don't just do computers, and to appeal to the part of us all when this type of exploring and learning was new and fresh, *before we called it hacking*. I also wanted to represent the lot of us that, even though we mean well in our explorations, are prone to causing trouble.

This story could mirror computers in the sense of it exposing user information and security issues: I had gotten a new tape recorder with a mic input and level control. I took the Pioneer headphones and attached them to the mic input, put another set of headphones in my ears, pressed record, play, and pause all together so that I could hear what was happening without wasting tape space, and turned the mic level and headphone volume all the way up. I then took the over the ear cup headphone acting as the mic and placed it next to the number dial on my locker at school, spun the dial three times to the right then slowly left until I hit the first number, studying the sound it made - a minute "tick" - made when you just miss the number, not stop on it. I thought of this idea after seeing one of the locks disassembled and studying how it worked. Once I knew the sound I wanted, I moved on to the locker next to mine and within a few minutes had it open.

You may be wondering how I had the chance to do this and also have perfectly quiet conditions in a school. My locker was right across the hall from the room used for detention. When we are all dismissed, the small hallway in a remote corner of the building stays lonely until the next day. Most of my friends and I got detention on purpose just so we could all hang out together after school and roam the grounds.

Anyway, I went back a bunch of times and wrote down the combinations to all the lockers in that little hallway, about 40 or so. Then I went back with a tagging marker and wrote all of the combinations on each of the lockers, luckily never getting caught (I kept my locker empty of contents and wrote its combination as well - using another locker upstairs to keep my stuff in). I witnessed the custodians installing new locks the next week. They installed a patch without addressing the underlying cause of the issue, leaving it open to a repeat. The motivation for all this ridiculous work was that someone had gone in my locker and stolen my Starter jacket, and also because I just wanted to see if I could get into all of them!

One more! Our favorite: social engineering. My home life until school ended was a never-ending atrocity - yes, in the rich quaint neighborhood.

(Kids: don't do what I am about to write, especially now that cameras are small enough to fit in your pee hole and located every three feet to capture everything everyone is doing. You're also more likely to face legal recourse for doing what you're about to read in this era of paranoid security. If you end up in the Feds, there is no room to maneuver out of doing time.)

Sometimes I just needed to not be home - or at school. My neighbor and BFF Vas500 had an alarm on his house. Of course, they wouldn't tell me the code, so I wrote down the alarm company's phone number (conveniently written on a post-it next to their phone) for later reference. One day at school, I decided I wanted to cut half the day. So I

went to the phone in the cafeteria (without Rich's box), called the alarm company, told the woman that answered that I'm calling from school, about to have an early dismissal, that I couldn't reach my mom, and that I don't remember the alarm code. She asked me my full name and birthday, along with a million other questions - so I gave my friend's information. She gave me a four-digit code and also the word that you use on the phone with the alarm company when they call you in the event that you'd accidentally set the alarm off yourself while at home. Social engineered!

My friend kept his house key in his backpack in his locker, which I remembered the combination to after looking over his shoulder one day as he opened it (yet I did not ever see him entering the alarm code!). I got the key and walked the half hour home, opened the door to his house (because I knew when his parents were not home), turned off the alarm, made a can of Chef Boyardee, and watched TV up in his room. When he returned home, I timely opened the door to greet him. Because obviously I'm a moron, I did not understand how he could have been pissed. He did not tell his parents about this, but I got in serious trouble with them anyway. You can't wait to know how.

The Chef Boyardee can that I ate I rinsed and put in the garbage. The bowl I used was washed and returned. However, the can should have gone in the brown paper bag that his mother (meticulous with everything) set up for recycling. She came home and found it in the wrong bag - and then out it came about what happened. Because I did not ever tell my friend that I had also made food, he must have been annoyed enough to spill the beans. You see? One little mistake and it's in trouble you get.

I obviously never got that because I'm writing this from jail years later fighting a case that, for once, I'm not actually guilty of (inmates are some true hackers - article in the works). I could fill a large book with all the stories of things I did and got away with. There's a rebel in all of us. The line between hacker and mischief for me is a slight shade of gray. More times than not, I'm just trying to get into or at something just to see if I can do it, but I have a bad habit of trying to get away with too much.

It's important that some of you - the ones who look in these pages because you want free money, want revenge on someone's Facebook account, or ways to break laws in secret - read this perspective. When you cross lines for personal gain - i.e., break into things to get money, someone's identity, etc. - you are then a criminal. I just happen to be a hacker with criminal tendencies. I'm not purposely trying to break any laws.

If we all were able to be conscious of our ability to resourcefully alter the use of things to accomplish a task of any kind, socially or physically, we would all recognize our own capabilities as born hackers, and perceive difficulty and adversity as challenges instead of excuses. As a direct byproduct, the misperceptions of hackers as represented by news and news publications, politicians, and victims of identity theft alike, would be replaced with a knowledgeable differentiation between *criminal with hacker capabilities* and *hacker.*

There should be groups one could attend once or more, even in schools, where qualities of a hacker are revealed and nurtured within the attendees who otherwise have no experience. By citing stories for a group, as done here, it would trigger stories of their own, their brains automatically parsing their history to locate a relevant experience to label "qualifies as a hack." Then, by drawing the inference to creative problem solving - not problem causing - skills, this would unveil a revelation in that person which would facilitate the building of a meaningful personal view of one's self as a hacker, an effect that is automatically positive and forwards a desire to explore life with this "new" gift of capability. By drawing out something in someone that gives them an opportunity to parade it to other people, you create fuel for them to advance their self-image.

One could worry that with such an influx of new "hackers," the word "hacker" would be synonymous with "hipster" in its overuse, popularized by an over-saturated field of inexperienced individuals romping around under a false pretense of the sobriquet. But leave judgment and elitism aside and clean up the scene first, no? Now we have only the definition of hacker as criminal to outsiders. To those in the know, even they are unaware that one in four hackers is a snitch That's a federal statistic from case law!

We all feel we lose something of our exclusivity when too many people like or do the same things as us, especially if they start term dropping, like when you're at a show and groups of people are just band name dropping. I hate that.

What other way to gain people an understanding of who and what we all are then if not to draw them into our world; inviting them to the possibility of discovering their inherent and creative flow as born hackers?

Hacker -n. - a person who possesses and uses instinctively creative and unorthodox means to both explore his surroundings and the contents therein, and improve upon them.

Explore. Investigate. Learn. Improve. Repeat.

Hack the future! Skal!

WYSE Moves

by Maven

I came upon the WYSE boxes whilst having to work supporting them. They are produced by Dell for various purposes and clients, including governments, large financial companies, and militaries within and outside 'Murica. This is an overview of notes taken from a predominantly passive experience of the devices, and some research undertaken afterwards.

This is only a short article detailing common defaults for WYSE Xenith boxes, principally version 8.0_306 WYSE ThinOS firmware.

These boxes are designed to act as thin clients, and front links to Citrix ICA servers, using XenApp or Xendesktop, for example.

Reading the online manuals (such as `https://www.rm.com/_RMVirtual➥/Media/Downloads/wyse-xenith➥-administrators-guide.pdf`) will tell you the following things, almost all of which are left activated on available systems:

1) Unplug the network cable prior to boot. This leaves it in a suspended mode through which you can perform items 2 and 3.

2) The network settings are editable, providing you perform item 1.

3) You can force a reset if you perform item 1, if you can't normally. That is, you can command the unit to go back to factory settings after reboot. To do this, select reboot, and a check box appears to "Reset to Factory Defaults on Reboot."

Before I carry on the list, all the settings are controlled by an INI file called "wnos.ini" that is loaded from the server. This file controls all the actions and permissions that the particular client has, and this file trumps the one that is cached locally on the DRAM. Whilst wnos.ini contains the global settings, there is also the possibility of "{username}.INI" files for more finely grained control of user access.

This is supposed to make them more secure, but nothing says that spoofing this file is impossible - they are easily generated by tools such as "WYSE WNOS.INI Configuration Generator" located here: `http://michaelkindred.➥wordpress.com/2012/03/28/wyse➥-wnos-ini-configuration-➥utility/`.

Let's continue the list of things that are possible:

4) You can easily view the current INI file on the DRAM and see its settings through the System Information link, which is normally available to most users.

5) If you are reset to the defaults, then pressing Del or Shift during boot will bring up the BIOS screen. It will ask for a password, which is case sensitive. By default it is "Fireport".

6) There is a G-Key reset "feature." Here, you tap the "G" key during boot, if it is not restricted in the cached INI file or the device has not been reset.

7) If you can access the file store, then there is an "Include=$mac.ini" - and if the mac.ini file has an "Exit" option. If you set "Exit=yes" then the file will return to wnos.ini. If you set it to "=all" then loading the rest of wnos.ini is ignored - this means that the protection of *all* relevant INI files should be ensured to prevent manipulation of security parameter loading. Remember: programmers put their includes first. They are in the /wnos/inc/ directory.

Based on the manual mentioned in the URL above, the boot process checks for wnos.ini over an ftp connection. If you were to use a dropbox such as PwnPlug, pre-loaded with an INI file created using the tool referred to above, then you could theoretically force a client to boot with different options in the following way:

I) Disconnect network adapter form WYSE box.

II) Edit "Network Settings" to point to custom FTP source

III) Allow client to boot using this INI file

IV) Reboot device

V) So long as the WNOS file is of a newer version, it will supersede older versions of the file.

Let it be said that there is no signature checking deployed in this process explicitly, although without testing, this is only a theoretical attack vector. Let it be said that the same is done for the BIOS image files, called "xpress.rom" - the process of reverse engineering the ROM might be tricky, and there are plenty of bugs on ThinOS.

The list of things that are controlled by the INI files is long. It includes some baby scripting options, as well as the following options: $MAC (MAC address - very unsure how this is used in communications), $IP, $UN/$PW (username and password used for sign-on), $DelCertificate=all (this would delete all certificates), and VncPassword (readable in the INI file viewer).

There are many problems with ThinOS. As it's a front for ICA, the handling of windows is very primitive - like, Windows 95 primitive. It is not unknown for windows to be displayed, read only, behind the "Locked Screen" login prompt, especially if these windows are running clients that make persistent connections. If someone had been looking at sensitive information (personal data, for example), then this can be viewed by all.

They are worth attacking due to some zero-days that are still likely in the system, despite being reported to Dell through official channels. They both have to do with the proprietary Autoshutdown routine not being able to cope with persistent connections held open over an ICA connection. Given how important the network connection is to the architecture of this WYSE system, such connections that are in common usage - connected to database clients and mainframes, etc. - should be handled properly.

The zero-day is as follows: Wait just under two hours. The system will start to autoshutdown. The screen will wake up and the screensaver will come on. It will display the user's session, with a "cancel shutdown" box. This box will start looping, and so keep the session alive. Move the mouse, view the screen, and click cancel. You are now in that user's session, as the autoshutdown has the ability to defeat screen locks but not persistent connections, with disastrous consequences.

Lastly, note that ping and traceroute are available to all users by default. Ping in particular supports DNS resolution of external clients behind the firewall, and can be used to check just this property of the network for, say, DNS tunneling.

This is the limit of our research. We haven't managed to get ahold of a unit for firmware reversing or the like, primarily owing to other projects and time. If others have access to such hardware, go ahead. I am sure there will be much to discover.

Manuals that are more up to date and official (that also, notably, have the default BIOS password removed) can be found at `http://www.wyse.com/manuals`.

Office Talk or Social Engineering?

by Gregory Porter
backfromthemovies.blogspot.com
greg.e.porter@gmail.com

Office jargon is a form of social engineering. It exists to maximize employee efficiency by making employees happier, or rather by making them think they are happier. But before expanding the previous statements, we ought to define both social engineering and office jargon.

Consider, social-engineer.org's definition that it is "the act of influencing a person to accomplish goals that may or may not be in [the target's] best interest."[1] Although there is a tendency to connect social engineering with computer security, such a connection is not a restriction. The emphasis of the above definition is persuasion, which is something that is present in all facets of society.

"Office Jargon" has many names: corporate lingo, corporate speak, business speak, commercialese. If you have experience in Corporate America, you may be familiar with such terms, but if you aren't familiar with these terms, consider the following examples.[2]

Synergy - Effect of working together

Touch base/reach out - Meet up with/contact a colleague to discuss progress

Leverage - Utilize, make use of a resource

One might hear such terms in emails or conversations. "If you need anything just reach out to X or Y. They are great resources; leverage them. Let's touch base tomorrow but, in the meantime, do what you can to promote synergy."

To trace this system of manipulation, we should start with Frederick Taylor, the founder of "Scientific Management" (aka "Taylorism"). Although the terms are treated as synonymous,

"Taylorism," in the classical sense, was replaced by the larger field of "Scientific Management."[3] The general idea of this research was to maximize the efficiency of factory workers. Tasks were broken into successively smaller discrete parts, whereupon anything unnecessary was removed.

This style of optimization focused on the efficiency of the system or process itself, rather than the inter-workings of that system (i.e., the human laborers). The laborers in this equation are seen as little more than cogs in a machine; once given a task, workers are expected to carry out tasks as commanded. The worker is not to modify the system or even provide suggestions. Workers in some cases revolted as a result of this type of mechanization.

In 1924, George Elton Mayo and his team were conducting research in a Chicago factory.[4] They were trying to determine what light bulb brightness resulted in the highest efficiency. They concluded that it wasn't so much the physical environment, but the emotional and psychological state of the workers that determined efficiency. While the study has been criticized, it marked a departure from, up to that point, the coldly scientific managerial experimentation.

It wasn't until the 1950s that researchers at Carnegie Mellon and MIT began to articulate various theories about management. It was out of this research that the corporate vocabulary as we know it was born. The most popular theory behind the terminology was that workers were thought to be "ambitious self-motivators who thrive in an atmosphere of trust." Office speak was considered to be a way to create an atmosphere of trust. Out of this Carnegie Mellon and MIT research, this new vocabulary nested itself in the corporate environment, particularly in human resources and in marketing departments. Given the nature of office speak, this isn't surprising. Marketing is inherently focused on presentation while human resources focuses on, well, the human resources of a company.

"In a workplace that's fundamentally indifferent to your life and its meaning, office speak can help you figure out how you relate to your work - and how your work defines who you are," concludes Emma Green. These words or phrases exist to alter your perception of the workplace to boost your efficiency. It isn't about making you happy, but about making you think you are happy (or at least happy enough to work). Linguistics professor Geoffrey Nunberg notes, "You can get people to think it's nonsense - at the same time that you buy into it." Indeed, Jack Welch (former CEO of General Electric) wrote that such corporate management systems would create "a company where jargon and double-talk are ridiculed and candor is demanded." Matthew Stewart provides several, now-universal euphemisms for firing people: streamline, restructure, let go, create operational efficiencies.[4]

Now, the elements of social engineering in this language are self-evident, but the relationship between Engineer and Target is complex. It isn't simply that your manager is employing the language to manipulate you for his or her gain. That might be part of it, even if it isn't a conscious decision. But such a relationship runs far up the corporate hierarchy. Your manager's manager speaks the same way. Even your coworkers may speak to you that way. Everyone is trying to find the most effective form of communication, but why? It is the bottom line of the company as a whole that drives us toward efficiency, which includes altering our language.

How do we, readers of *2600*, protect ourselves from such language? But before even asking that, are we so sure we want to or have to? After all, between "you're fired," or "you are being let go," which would you rather hear? The "truth" that you are losing your job, or that the company is more efficient without you, is present in both phrases, but the latter is easier on the ears. Perhaps, then, "protect ourselves" is too strong a phrase. We must at least be aware of such language. Instead of saying "I will contact X and Y for help," we are saying, "I will reach out to X and leverage Y." These are words that were carefully chosen to exploit positive connotations for the benefit of the company. It is a vocabulary that manages to dehumanize with a smile.

Sources

[1] "The Official Social Engineering Portal - Security Through Education," *Security Through Education*.

[2] Weiker, William. "What Your Boss Meant to Say," *Dictionary of Management Jargon*. William Weiker.

[3] Drury, Horace Bookwalter (1915), *Scientific Management: A History and Criticism*. New York, NY, USA: Columbia University.

[4] Green, Emma. "The Origins of Office Speak," *The Atlantic*. Atlantic Media Company, 24 Apr. 2014.

Archiving COMIXOLOGY™

by Ook

Comixology is a neat site where you can buy comics online and read them. However, I'm not always in a position where I can browse a website, so I like to archive the things I buy. Comixology doesn't really permit you to do this, so I felt I should see what I could do to permit myself - after all, I can see the images displayed on my screen, so they're in the clear somewhere.

My first attempt to pull things off a website is the same as anyone else: look at the source (unhelpful), then look at the network traffic (just as unhelpful, but interesting). The images don't exist as-is on the pipe anywhere - however, they do exist, as image scrambles with no simple pattern.

Next, I decided to look at the DOM (Document Object Model) to see if the page script itself was assembling the images in some coherent way - it was! The pages were composed of a seemingly random number of canvas tags. So, the simplest attack would have been to use the toDataURL method on - oho! toDataURL was set to null! This was simple enough to restore (recently I'd looked into sandboxing localStorage away from a potential attacker via the same means, only to find it functionally impossible). Now that I had a data URL, I saw some very interesting stuff: the canvas images each contained a subset of the comic page in random staggers. To get the full image, I'd need to copy the canvas data to a new canvas. A really neat perk, though, was that they were creating the canvas at full resolution, then using CSS transforms to scale it down. I could get full resolution images!

Teaching a script how to click the "next" button and wait for the DOM changes that would signal the next page had loaded was easy enough. Later I worked out that I could just trawl the page selector at the bottom - both to get the full set of pages and to choose each one.

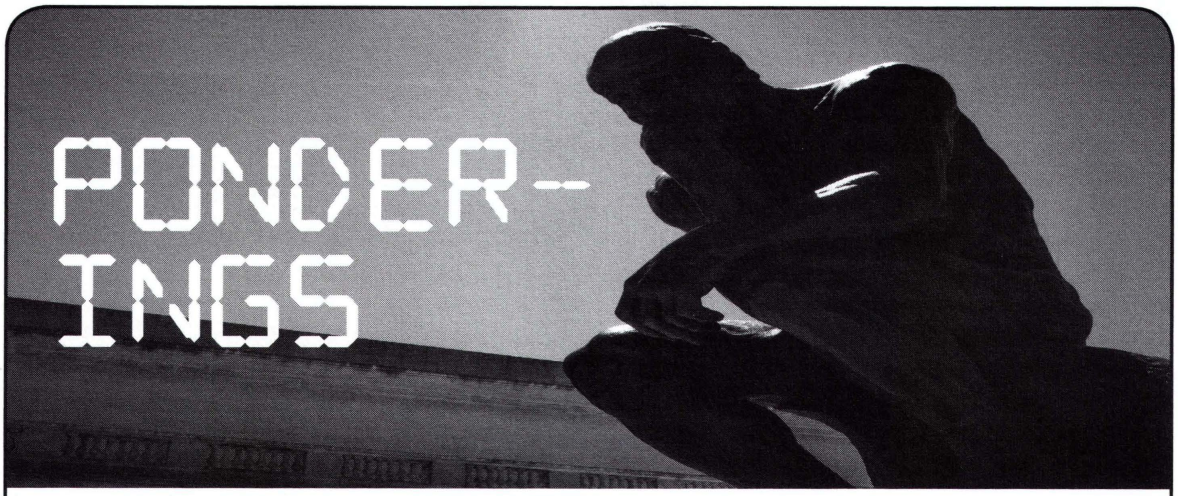
I was able to use Chrome's FileSystem API to then save the composited images individually, but getting them back out was painful. Even with eligrey's useful FileSaver, I'd get a bunch of jpg or png files - that's neat, but there had to be something that would be more "click a link, get a file."

Using stuk's JSZip library, I found I could create a zip file in memory within the browser - I could just create a CBZ file!

I have a friend who's really into comics as well, and figured he might want to be able to archive his stuff. So I built a small UI to let him select the quality of the downloaded CBZ (especially for longer comics; full resolution PNGs were averaging five megabytes a page, and a particular 165 page comic was crashing Chrome when attempting to build a CBZ file of almost a gig in size).

The finished, commented code is too lengthy to print here, but is available on the *2600* Code Repository (`http://www.2600.com/code/`). I share with a warning: They put quite a bit of work into preventing theft: encrypting the image data, shuffling it, splitting it between canvases, obfuscating their code, etc. I didn't do any kind of analysis to see if they were embedding compression-resistant steganographic watermarks in the images to concretely identify me as the purchaser should my archives get out into the wild - but if I were the programmer on the project, it's something I'd have recommended to enable suit should my copyright be threatened by unchecked file sharing.

Don't help others steal things - but if you do, analyze the images to make sure it's not traceable to you as well.



Opportunities

Dear *2600*:

I am David Wei. I am involved with the Guiyang intellectual property bureau which is in the Guizhou province of mainland China. I am getting in touch with you regarding property investment that was facilitated by myself and my colleagues a few years ago.

We had started this process with a gentleman by the name of Mr. Norman Gerr a while back but had to suspend same due to unfortunate events concerning Mr. Norman. I would respectfully request that you keep the contents of this mail confidential and respect the integrity of the information you come by as a result of this mail. I contact you independently and no one is informed of this communication.

We contact you however because you share a similar surname with Norman, please get back to me once you get this letter regardless of being related to Mr. Norman in anyway as this can be very beneficial for all involved.

I await your response.

David Wei

Sure, we sometimes print spam, only because it's unique, funny, or perplexing. So the letters department has a similar surname to Mr. Norman Gerr? But apparently that doesn't even matter since these people want us to contact them regardless of whether or not we have any relationship to this guy. And just what "unfortunate events" befell poor old Mr. Norman in the first place? If only we had the time to thoroughly investigate each of these little stories. What's particularly intriguing here is the fact that the scam isn't leaping out at us. Usually, there's a request for login info at a thinly veiled fake domain, or a .zip or .exe file we need to open right away, or even simple banking info so our account can be pilfered. In this case, there was none of that, just a request to write back (to an email address that is in Mexico for some reason). Perhaps the scam begins in the second act. Regardless, this could be the start of something truly amazing. After all, it's what Mr. Norman would have wanted.

Dear *2600*:

I am happy!

:-)

Happy Man
<happy@kaundaemail.info>

We're pleased to hear this, but again we're wondering what the angle is here. There's no attachment and no instructions to do something that will wind up hurting someone or infecting their computer. Could this be someone who is just genuinely happy? We hope so. Enjoy all of the email you will soon be flooded with.

Dear *2600*:

I have written various web security articles on my blog that I would love to see published in *2600* with edits where appropriate.

Amer

We'd love to publish them, but we must insist that any articles submitted not appear online or in another publication until after they've been printed here. We support recycling, but not that kind.

Dear *2600*:

I am a technical writer with over ten years experience in the Information Technology sector.

My background is in the real-time deployment, administration, backup, print server administration, and multimedia authoring for heterogeneous client/server local area networks based on the Original Equipment Manufacturer resale model.

I am interested in submitting articles based on for publication with respect to: "2600:Magazine." based on the following content:

Christopher

We're going to stop you right there as we don't think our readers need to see the next 14 paragraphs (we're not kidding). Something tells us you've never actually read a copy of our magazine. If so, you would know that we're not anywhere near as formal as this, and that our language tends to invoke a lot more excitement than what you've given us here. For example, you go on to

say: "Based on the relevance of the technocratic corporate policy and corporate governance of the marketspace defined as the information technology market sector with respect to quarterly earnings summaries and the corporate vision for the market product line." That one sentence will put children to sleep almost every time. In fact, it's so dull that it took several readings before one of us noticed that it actually isn't a sentence at all, but merely an enormous phrase - meaning it will need even more words to get to a point.

We don't mean to be overly harsh, but it is rather enjoyable, particularly since we believe you've probably sent this same identical request to a number of publications. Let this serve as an example to prospective writers of what we don't want, either in presentation or in content. There is so much of excitement to cover in the hacker world, from history to new technology to mischief to legalities. We find that most of our writers put together pieces where you want to keep reading to see what happens, not simply to get it over with.

Dear *2600*:

I'm a special education teacher who attended the second HOPE conference many years back and who might be in a position to teach cybersecurity and cryptography to kids in New York City's public school system. I'll explain how, and I'm also writing for possible assistance. Last summer, I enrolled in a summer workshop for teachers at NYU Polytechnic that focuses on robotics and mechatronics. While there, I also found that NYU Poly offers a similar summer workshop for teachers focusing on cybersecurity. The teachers in that program learn the basics, and then help to design lessons that teach white hat hacking to their students. Although I plan on hopefully returning to mechatronics this summer, there is the outside chance that I may well enroll in the university's ethical hacking program instead. If I'm selected, I'd then be expected to help teach coding to my kids at our elementary public school, something I've already taken steps toward.

I'd like to propose a partnership between *2600* and NYU Polytechnic. Since I would teach ethical hacking and cryptography, it would be great if someone from the magazine or one of your HOPE conferences would consider guest lecturing at my school, either by Skype or in-person. It would require no more than an hour that would take place at that person's convenience, and the more people who wish to get involved, the better. The goal would be to produce a new generation of hackers, which would mean more *2600* readers and HOPE attendees.

Lee

This could be an interesting project, if done properly. We've always had a dim view of the term "ethical hacking," but it would be foolish to get caught up in semantics. Of far more importance is the ability to reach people in their formative years to hopefully steer them away from the many misperceptions that are aimed at us through the mass media. We can think of nothing more healthy than kids learning how to use encryption to protect their privacy from individuals and institutions that seek to take advantage of them. If that is the aim here, then we support the idea.

Unfortunately, we ourselves cannot commit to much more than this, but this is why we have our monthly meetings throughout the world - so that people who get what we're all about can connect, exchange ideas, and embark on projects just like this - and hopefully do it right. We suggest heading over to the New York meeting on the first Friday of the month and meeting some of the people there who are more than qualified to work on this. The same holds true for similar projects (and very different ones) in other cities. The hacker world is filled with amazing and inspirational people. We hope to hear good things about how this has turned out.

Dear *2600*:

Just confirming, we have dinner reservations at 10PM?

Timothy Castro
<email@e.advenze.in>

Timothy, we're so sorry the entire editorial department here stood you up, but we lost track of time while trying to figure out the angle of whatever scam this one happens to be. No attachment, no website to go to, but a really snazzy email address. Perhaps we're supposed to go to that site in a browser? But that would only pull in those people who were curious like we are. All we know is that these weird emails are keeping us up at night and preventing us from getting any actual work done. Well played, NSA.

Dear *2600*:

I have done some alpha and beta testing of games and some software/hardware beta testing. I would like to put some of those experiences down in an article and submit. Maybe even a separate one on my Wal-Mart experience (not one hundred percent hacking, but Wal-Mart pays me to talk about improvements to their store to them).

J

These all sound like great ideas to us. We're waiting by the email box.

Idealism

Dear *2600*:

In the hypothetical world of whistleblowing; if an individual wanted to anonymously send a company-wide email blowing the whistle on wrongdoing and mismanagement, how would

(s)he go about that without spam filters blocking several hundred emails coming from the same source? It's like the movie *Jerry Maguire,* but (s) he doesn't want to be a martyr. Thanks.

Tom Cruise

It really depends on how the spam filters are set up. You might try testing them out first with something that doesn't draw much attention, but that isn't obvious spam. If your account gets a copy, then odds are everyone else did as well. If that doesn't work, perhaps not sending them all at once would be the answer, assuming you had to use the same account to send from in the first place. One other option you might want to consider is to simply create a website with an easy-to-remember name and have word of that site leaked in various ways to employees. That way it doesn't matter what defenses are in place - the info is someplace else out of their control. Obviously, we assume you've got the basics down insofar as covering your ass. IPs are often revealed in emails and domain registrations can be uncovered as well. Good luck with your mission.

Responses

Dear *2600*:

Thanks to Kevin for his article in 31:3: "Forensic Bioinformatics Hacks." I remember hearing about the article retractions that resulted from your analysis, but never heard the inside story of how the errors were uncovered. Your article was a fantastic example of the value of publishing scientific data, and the need for also publishing (and vetting!) the code.

My own scientific dataset wrangling often involves ad-hoc creation and destruction of spreadsheets, arbitrary sequences of grep/cat/cut/sed/awk/etc., and other hard-to-replicate processes. So, perhaps it's impractical to have all phases of software for data analysis be submitted with an article. The real "programs" though - whether MATLAB or C or whatever - should be easy enough to capture and provide.

I'll share your outcomes with the bioinformaticians at my workplace and elsewhere, so they can better understand the value of correct and replicable programming. Plus, of course, the benefits of diligently following the discovery paths taken by colleagues and predecessors.

It's remarkable to me that modern use of computers has resulted in less replication and examination of base assumptions than by prior generations of scientists and engineers. Reverse engineering the analysis shouldn't be necessary to provide one of the most fundamental requirements of science: replicability.

Estragon

Dear *2600*:

I, for one, would be all for having a CD-ROM subscription of the digest PDFs and back issues. I mean, they used to have CD-ROMs of Usenet years ago, so why not? Bonus points if it's a pressed disk like Wolverine Bates suggested. Downloads are fine if you have the bandwidth and keep backups (and you *do* keep backups, don't you?), but they can't match the availability and reliability of a physical copy.

Anybody else? Let's make this happen!

Mistman the Magnificent

Dear *2600*:

In the January 6, 2015 issue, "sueicloud" wrote a letter about using a MAC address to obtain "unlimited" free Wi-Fi in hour-long increments. Your response was reasonable, but I noticed an interesting detail hidden in that article.

As one of the "suckers" who pays for service, I did some research on Comcast's FAQs and found out that, by default, Xfinity's current line of routers/modems ship with a somewhat hidden feature enabled - similar to a guest account. This basically turns your own home router into the Wi-Fi hotspot described in sueicloud's letter.

Granted, your own home network should be isolated from those on the guest network - as long as you trust that Comcast did not leave any security holes in their firmware (or don't worry about the zero-day vulnerabilities that eventually get discovered!).

But for those of us who would rather not share our Wi-Fi bandwidth with the world, this feature can be disabled at your account on Comcast's website.

On the surface, this seems like an innocuous feature - Comcast is simply trying to create a network of Wi-Fi hotspots across their service areas, which is certainly an added benefit for their customers. Anywhere you see an "xfinitiwifi" AP, one can use their own credentials and get wireless Internet. However, this feature looks a bit darker when you discover that they are implementing this by turning their customers into unwitting hotspot providers whenever they install a gateway/router and hiding this information in the mountains of fine print you get for signing up.

Neil N.

That is indeed fascinating. We wonder if other cable companies do the same thing without really telling their customers. (We also find it interesting that digital subscribers often refer to an issue of our magazine by the date it shows up on their device rather than the season or issue number.)

Dear *2600*:

Kudos to *2600* for printing photos of the Malaysian payphones (and to Bryan Rhodes for somehow taking them)! I never cease to be amazed at the technological wonders and sky-high aspira-

tions of tele-conglomerates. I mean, wow! A payphone in the exosphere! The Soviets beat us with Sputnik, and then to rub it in, Malaysia goes and puts a payphone up. This is why we need to fund NASA, people!

ghostguard

Dear *2600*:

I am a US Bank customer and I login to the website using Linux. So I am not sure why "A Friend of Freedom In Cottage Grove" (34:4) says they don't support Linux. He provided a link to the US Bank login help page, so I'm guessing that is the place to go if anyone has problems logging in regardless of the O/S platform in use (or just call the bank - I've always found the US Bank to be reasonably helpful with any online banking issues).

David

Dear *2600*:

The file concatenation trick described in the article "Taking Your Work Home After Work" (31:4) by GerbilByte is one of my favorite tricks in circumventing the file attachment restrictions on my employer's email system. They do matching on file types, based on the file extension, and js files are blocked, even if inside zip files. So we use file concatenation of a jpeg and a zip - I used a picture of a mule, which seems appropriate - it being a mule in more than one sense of the word.

To unpack the files, we have a simpler method than that described in the article - use winrar and open the jpeg. Sounds weird, I know, but winrar looks at the file and recognizes the zip file content and shows that. Then it is easy to extract the zipped files.

Rob

Dear *2600*:

I am a regular *2600* Kindle edition reader/subscriber/fan. I read every issue from virtual front to back (even though I admit sometimes I skim sections that are beyond my comprehension, especially the technical programming points). However, I wanted to make a couple of comments from 31:4 (digital edition).

First, I had to do a double-take when I noticed the header throughout showed up as January 1, 1970! Oops!

Second, and more substantially, I want to challenge a small but important point arising from the editorial in 31:4. There you say, "It seems as if anyone believes they can now be a filmmaker. But of course, not everyone *is* a filmmaker. Just as not everyone on Flickr is a photographer, not everyone who has a blog is a writer, etc." You then go on to say that ease of access to these online venues does not equal quality of contribution. Granted.

But my question is: What is the point at which someone actually *becomes* a photographer, writer, or filmmaker? Since when does quality of contribution constitute whether one is actually engaged in those activities? Your way of putting it might seem common-sensical: just because someone can put a video online doesn't mean it will get a million hits or win a Pulitzer or Oscar. That is so obvious that it barely goes without saying. Internet utopianists who believe that somehow the Internet has or will put everyone on equal level are increasingly being shown to be wrong. Inequities exist on the Internet as they do everywhere else.

That said, I still ask, *when* is it that a person becomes a filmmaker or writer or photographer? Does the number of views or readers constitute when a person *becomes* a photographer or writer? I refuse to allow that to be the criterion. No, it is not the size of an audience that matters, but the action of the creator that constitutes a writer, a photographer, a coder *as* a writer, photographer, and coder. Even a Flickr photograph viewed by no one else other than the creator herself, or a crappy blog post read by no one but the author himself, or a small C++ executable that does nothing more than say the proverbial "Hello, world!" are all still products of someone who chose to do *something* rather than *nothing*. And the fact that "novices" may not produce something of the "quality" that will please the hordes does not negate the fact that these individuals actually bothered to get out of bed and *do* something!

Not everyone is a pro, true. But I would rather applaud the person who produces a cheesy YouTube video or writes a piece of code that does nothing more than flash random numbers on a screen than try to assure the flimsy self-esteem of the millions who spend their days doing nothing but consuming *Dr. Phil* or *Oprah* and who rarely attempt to learn anything new beyond how to fill their mouth full of potato chips with greater efficiency.

My perspective is as follows: To draw an invisible line between the filmmaker and non-filmmaker, between the writer and non-writer based entirely on quality - goes against the very spirit of hacking which I have discerned in the pages of *2600*. On the contrary, I've learned from *2600* that hacking means trying something new, learning a new skill, being inquisitive and taking a risk. The beginning blogger posts her or his first blog post often with trepidation because they so often assume it isn't "good enough." Of course it isn't - it isn't good enough to win a Pulitzer. But it is better than writing nothing at all. So what if no one reads it! I say to that person, then write some more and make it better and maybe next time someone will read it and be entertained, informed, or maybe even moved to action. But let's not stoop to the level of allowing some literary elite to say, "Well

he/she is obviously no writer."

Let me add another slightly different perspective. Doesn't "hacking" (at least the kind which *2600* wishes to promote) include with it a social dimension of "encouragement?" My 12-year-old daughter is a beginning photographer - and yes, she *is* a photographer because she has a camera and takes pictures! (Just as a blogger is a writer and a person who writes her or his first program is a coder.) Are all her pictures high quality and stunning? Hardly. Has she taken some pictures which I look at and go, "Wow! Cool!" or "What a different perspective!" Yes. I do not subscribe to the belief that we should tell kids that everything they do is excellent. That is obviously not true. But I do encourage her when she makes improvement or does something cool with her camera. And when she does, is that not actually a manifestation of the spirit of hacking itself?

As for me, I am a 47-year-old senior executive in higher education with a plethora of interests from writing to ham radio to electronics to coding to photography to exercise to astronomy to urban planning to mechanics to woodworking to... well, the list goes on. In most of these (with the exception of writing in which I am actually often paid to write), I am far from "professional." But I am a photographer, an astronomer, a woodworker nonetheless. I'm not trying to brag about the breadth of my interests and accomplishments (far from it; at best, I'm barely a novice in most of these areas). Nevertheless, I have benefited tremendously from actually trying to learn a bit more about all of these interests, to find better ways of doing things, to fix things instead of throwing them away, and to enjoy doing them and even occasionally have others enjoy the fruits of what I do as well. In that regard, "hacking" is not about being professional or non-professional; it is not about high or low quality; it is not about greater or lesser expertise; it isn't even necessarily about technology itself. Hacking is about trying something new, about learning from mistakes, about encouraging others in their successes or encouraging them to learn from their mistakes. Most importantly, I think, hacking worth its name is about contributing to the common good of our society as a whole, even if it does give greater joy to the one doing it.

One last thing - and I left this to last, lest I lose some readers too early because of bias or prejudice: I have a deep commitment to Jesus Christ and am a professional theologian. For many, that fact may negate everything I've already said, or somehow disqualify me from the conversation because they think I'm a religious nut. Whatever. I'm fine with that. To those who think I'm deluded or wonky, you are entitled to your opinion. You don't have to share my theological frame of reference for me to uphold your dignity and the fact that you have your own brain, your own opinions, and your free will to believe whatever it is you have chosen to believe. I simply say: Don't stop learning or exploring because in the end, the opposite of hacking is not a closed mind, but a mind that refuses to accept that it, too, has its own biases and prejudices and which thinks that the only truth is that which lines up with the present state of one's own brain. I know that many religious people are just like that: They equate the content of their brain with the truth, but the reality is such perspectives are found everywhere, in religious and non-religious people alike. Hackers, on the contrary, whether religious or not, at least admit that they could be wrong. But they also seek the truth with the conviction that it does in fact exist. If it didn't, what would be the point of any form of inquiry at all?

I end with a point that I doubt has ever been made in the pages of *2600*. Was it not Jesus Christ who taught us the golden rule: Do to others as you would have them do unto you? Perhaps not many would have thought of Jesus in this way, but I think (and this is an opinion only, not some kind of dogmatic statement) that Jesus the carpenter from Nazareth was probably a "hacker." Scripture says that he "grew in wisdom" and I think in part that even he learned how to do things better, not only for his own pleasure, but also for the good of others. And in the end, isn't that what "hacking" is all about? About not only learning and trying new things, but also encouraging others to do the same? And then to share in the joy of such discovery and growth?

Maybe I've made a mountain out of a molehill. As I read this over, I find there is so much more to say and that even my own argument may be weak or missing the point. If so, oh well... at least I enjoyed thinking this issue through in print and hopefully entertained or even caused someone, somewhere, to see or think about something in a different way. And if that is the case, then it was worth writing a letter to *2600* rather than writing nothing at all.

Saskman

First off, we applaud you for writing a thoughtful letter to us, especially as a digital subscriber. There are many who believe that the digital world is leading us down a path of anti-literacy and it's nice to see that disproved.

There is little you've written that we can honestly disagree with. We feel you may be taking our point in the Winter editorial a bit too literally. Yes, technically, anyone who can pick up a camera and take a picture is a photographer. But with virtually everyone now doing that with their phones for every inane bit of subject matter imaginable, there needs to be a way of defining true art from a mere

fad or an activity that has no passion behind it. Perhaps just inserting the word "good" or "decent" in front of the skill in question would serve that purpose. Our point was that so much is being drowned out with all of the noise out there and that it's really easy to become discouraged. What we're hoping for is that hackers, artists, and professionals of all sorts pursue their passions and not feel as if their goals are insurmountable because so many others seem to share them. Easy access to technology will open a lot of doors, but in the end it's those who stick with it who will contribute something significant. It doesn't happen easily or overnight, and often it takes a lot of trial and error. We appreciate your taking the time to make us think this over some more.

As for the 1970 header you saw, we have no idea what that could have been, but it didn't show up that way on any of our devices. If anyone else noticed any oddities, please let us know.

Dear *2600*:

I am sick of reading yet another article by lg0p89. This guy must submit a bunch of articles every quarter in hopes of getting published. Every time I see his name as the author, I know that I'm about to read yet another content mill worthy article. I suggest limiting authorship of a published article to every other issue so that more individual voices may be heard.

In order to help rectify the situation, I offer an article of my own. However, I currently cannot write it without serious jeopardy to my upcoming release from federal prison. In 29:3, an article on the TRULINCS computer system in the Bureau of Prisons was published. I developed an automated program which operated through the public messaging "email" system. I obtained a root shell to my own VPS with only the minimum approximately three hour delay. I followed the prison rules to the letter and officials were unable to sanction me. Unofficially, without due process, against policy, and in violation of my rights, my email access was removed. I have spent two years appealing, only to be subjected to lost paperwork, arbitrary denials, and stalling tactics. They won, as I'll be released before I can file in court, thus mooting the issue. I hope the readers look forward to my article on hacking the BoP.

P.S. I should have finished issue 31:2 before writing in to complain about lg0p89's prolific writing because on page 53 there is yet again another of his many e-how.com worthy articles. I am beginning to suspect that lg0p89 may actually be an article generating bot. Bots which write sports news articles exist, why not *2600* article writing bots?

Delicious Cake

We hope none of our authors are non-human, at least for now. The "every other issue" authorship idea is an interesting one which we'll look into. As always, we'd like to know what our readers think.

Dear *2600*:

I read about the Source Interlink issue and have purchased a lifetime subscription to avoid their bullshit and help keep *2600* going.

I've been reading your articles for longer than I care to admit. I have enclosed my cards and bookmarks for your staff and would appreciate any warm referrals. We small business owners need to stick together.

Russell Nomer
Information Security & Management
Advisory Services
www.russellnomer.com

Hopefully, this will result in many referrals. We thank you for your support.

Facts and Theories

Dear *2600*:

Want to know the real reason why Sony withdrew *The Interview*? The reliable rumor is that Sony caved in because those terrible "hackers" found documents that proved Sony was a corporate criminal! The docs showed Sony was guilty of cyberterrorist acts against torrent sites, private individuals, and other companies, especially Google! Sony was also involved in more serious federal crimes like illegal campaign donations, money laundering, and influence peddling!

Sony, in effect, decided it was better to look weak and to cave in than to suffer from a federal criminal investigation, an investigation that could result in both civil and criminal penalties, as well as risking a drastic drop in the value of their stock!

Sony's CEO reminds me of *The Godfather* movie when Marlon Brando says, "It was *just* a business decision." Yeah, right!

Jay Jay

That's some reliable rumor source you've got. So now that the film has been released after all, where is all of this evidence that was supposed to be released? And why would Sony have ever believed that they'd be safe by following these conditions? Our *reliable rumor source tells us that Sony lives in fear of bad press and initially withheld the film because they believed that would be the result, especially if all of the secret North Korean operatives began to blow up theaters in the States. When they began to realize that this scenario was more farfetched than the one presented in the film (and when websites like ours began to offer to take the heat for them by showing the film online), that's when the damage control pendulum began to swing the other way. We also believe this is*

why that massive hack, initially spun to show the world just how evil and dangerous hackers were, turned into an inconvenience that barely affected their bottom line. Once people started to ask a few questions as to how such a thing was possible in the first place, blame became a lot less important than repairing the company's image.

Dear *2600*:

I just got an envelope from a friend through USPS. She's a homeopathic practitioner and had sent me a few grains of a remedy for lingering aftereffects of the flu that's going around.

I was interested to observe that the envelope had been carefully pierced, from the back side, through a couple of layers of paper and into a tiny manilla envelope within that contained some small homeopathic grains. The pierce-holes are rough-edged, around 4 mm x 3 mm, with a sort of hanging chad. The small inner envelope was targeted. The holes did not continue forward through the front side of the envelope.

Is this a common thing, that some sort of probes are inserted into envelopes to check their contents? Big Brother is everywhere and I'm sick of it!

M.

Years ago, we might have said this was a paranoid theory. (Hopefully, we would have known better.) Today, it seems well within the realm of possibility. It also seems quite likely that the majority of people would support such a thing, "in the interests of safety." The only way to be sure is to repeat the scenario a number of times between different parts of the country using the exact same contents. Apart from driving the authorities crazy, we get to learn just what it is they're up to. At least some of it.

More on 2600 *Meetings*

Dear *2600*:

Last time I checked the *2600* meetings list, there was still a meeting in Trondheim, Norway. Is it possible to contact the person who last supplied details about this meeting through you?

Tim

This is only possible if the meeting has a website and has elected to put personal contact info up on it. We don't act as a go-between nor will we give out anyone's personal info. As meetings have no leaders, your best option is to simply show up and see who else is there. Since this particular meeting was discontinued a while back, you would also need someone to pick a place and start getting the word out.

Dear *2600*:

Could I some how be put in contact with someone from the Virginia Beach meetup? I showed up to the Pembroke Mall and could not find anyone. There isn't a food court in this mall and hasn't been in about two years. So I went to where the food court used to be with no luck.

Jim

First, let's be a little petty and get the terminology right. Meetup is a product. Meets are for track teams. What you're talking about is a meeting. And even that's not entirely right because meetings tend to have a lot more organization than what you'll find here. It's actually more of a gathering. But we like the word meeting more and it's what we've been using for more than a quarter century, so we'll stick with that. Now then, to answer your question, we're sorry to say that after hearing similar reports of a nonexistent meeting place and a lack of attendees, this meeting has been delisted. All is not lost, however. Since other people have been reporting the same thing, that means there are other people in the area who are still interested in going to the meetings. So if you or someone else were to find a decent location and start getting the word out, the meetings could very well come back to life in your area. We wish you luck and hope to get word of this in the future.

Dear *2600*:

Two people showed up today, but most locals still go to the local Makerspace.

Lou

Two people is admittedly a low turnout - in fact, it's the lowest possible turnout you can have while still using the word "meeting." But it's something. Makerspaces and hackerspaces are great places to learn and work on projects, but they are completely different from the monthly meetings, which are more about being out in public and meeting new people, sometimes even ensnaring them as they pass by. This is why we discourage meetings that take place in establishments that aren't out in the middle of a lot of unrelated activity. The monthly meetings are ways of finding and welcoming new people who may have never met a hacker in person before. This has worked well in so many places over the years, and it's proven quite essential in portraying what the hacker world is to the uninitiated, which often includes the media. In this particular case, we see that there are no activities taking place at the local space you mentioned for the first Friday of the month, so there really shouldn't be any difficult choices that need to be made.

Dear *2600*:

Hi, I'm interested in starting a meeting. Could you tell me what I need to do?

Memo

All of the details can be found at our meetings page at www.2600.com/meetings. The most important thing is to keep us in the loop as your meeting starts to come together. We only list meetings that have enough organization to ensure that at least a few people are showing up at the ap-

pointed location and that someone is able to email meetings@2600.com with updates.

Dear *2600*:

I recently bought an issue of *2600* and noticed that the meeting information for New Mexico is outdated. The Quelab Hacker/Makerspace has changed its address and the meeting times are Sundays at 7 pm, as this is when the facility is open to the public for "Hacknight." I'm not 100 percent certain that there isn't another *2600*-specific meeting on other days.

Nolan

This is exactly why meetings at these spaces can be problematic as they have their own schedules that don't always fit in with meeting days. As our meetings are always on the first Friday (first Thursday in Israel) of the month, having one on Sunday only for this location would needlessly complicate matters. The "Hacknight" activities have their own place and shouldn't be combined with what we do with the monthly meetings. That seems to be in synch with the way the space is run, as there is no mention of 2600 *meetings taking place there. If you restart the first Friday meetings, we'll be happy to relist them, although we do suggest having them in an open and public area.*

Dear *2600*:

This may be news or not, but the Plano, Texas *2600* (one city north of Dallas) is now attempting to call themselves the North Dallas *2600* group and the Dallas/Fort Worth *2600*. This is an issue and is a clear and deliberate attempt to discredit and draw attention away from the Dallas *2600* group, which has been clearly established locally (and mostly with you guys too, with some lapses from laziness) since the late 80s. Please have them represent themselves as Plano *2600* only, otherwise it creates issues.

Matthew

We don't know what kind of territorial issues you're having over there, but they're really not anything we have an interest in. When the Dallas meeting fell off the radar, the Plano meeting was listed as "Dallas (Plano):" as it's a suburb of Dallas and we prefer to list the name of a nearby large city when possible. When the Dallas meeting reestablished contact, it was listed as "Dallas:" and this other meeting was listed as "Plano:". They can say they're the Pluto meetings if they want, as long as they follow our meeting guidelines. They obviously have to tell people where they are and anyone paying attention will find out it's in Plano. We don't see how that discredits or pulls people away from your meeting. We suggest you find a way to live with this, as we're not interested in turf wars, especially not any that have our name in them.

Dear *2600*:

The Philly meetings are going well. Making recurring stops and having good chats. I enjoy the crew self-moderation. Lively dialogs about really anything.

Meetings are a lively way to get out on Friday nights. If you are out, make it a social night with new friends. If someone troubles you, it is OK to not talk to them. This is the world and everyone is not for you. Use your own judgment and have fun. It should be pretty easy and natural. Give it a gander.

Pic0o

Dear *2600*:

I was hoping to restart the *2600* meetings in Scotland, particularly the ones in Glasgow. However, I remember there often being people who commuted to Glasgow from Edinburgh. Would it be OK to have a monthly switch meeting from Glasgow to Edinburgh and back? And could this be reflected on the meeting page?

TheGeek

This sounds like it would be unnecessarily complicated. We don't know if you're proposing having two meetings a month, alternating months between two cities, or having the meeting on a train going back and forth. Regardless, it's certain to confuse people. There will always be those for whom the first Friday arrangement is inconvenient, as well as some who aren't able to make it to the location. But if there are enough people who are able to work it out, there's no reason not to go ahead and have them. Both Glasgow and Edinburgh are big enough cities that are enough of a distance away from each other where meetings could exist in each of them. We suggest you focus on getting Glasgow going and then hopefully you'll find someone who can help build up Edinburgh. You should be able to find hordes of Scottish hackers. We look forward to hearing all about it.

Issues

Dear *2600*:

My two Facebook pages have been stolen.

Facebook has a serious security problem and a deceiving lack of care for its users.

I have worked four years to obtain respectively 113,000 and 138,000 likes on two Facebook pages to support my two websites: www.petyourdog.com (online since 2002) and www.kuromanga.com (a project in development).

The thief is presently using those pages and he is posting lots of garbage that has nothing to do with dogs or manga. This is ruining the image of my sites, especially petyourdog.com that has a solid reputation for 12 years and is one of the major resources for dogs on the net.

Facebook has obviously lots of care for the many billions of dollars they are making each year, but not too much for its users.

There is literally no means or ways of contacting anybody at Facebook.

They do not have any phone number whatsoever and their help is a big maze of filtering that basically says, "Do not bother us with your problems."

The best answer I found on the Facebook site is "please contact one of the administrators of the page to get your admin privilege back." The major problem I have with that answer is that the existing and only administrator of my two pages is a criminal and a thief. I would doubt he is going to kindly give me my pages back.

I have been working on those projects for over ten years now and there is no way I am going to let this keep happening. The only solution I presently have is to get their attention through the media.

My Facebook pages are: www.facebook.com/petyourdog and www.facebook.com/KuroManga. Facebook makes tons of money with their users. On top of that, I was a good customer for them, helping them to advertise. I cannot even send them an email concerning my problem. This is outrageous!

Richer Dumais

We have to admit that we initially felt compelled to write a very sarcastic reply to this problem as it starts off sounding pretty absurd. We would have said things like: Is this really what you spend your time worrying about? Or: You actually "worked" for four years to collect nearly a quarter of a million "likes" and you can say such a thing seriously in a sentence?

But then we realized that this is how a lot of people spread the word about their projects and businesses, in addition to their lives. And perhaps now we can all see that nothing comes without a price, especially when it's handed out for free.

One important detail we feel you should have included is just how these pages were taken over by somebody else. Knowing what the weakness was (easy password, stolen list of subscribers, security hole at Facebook, etc.) would undoubtedly help many others.

We had similar challenges finding a working phone number that actually connected to a human who could help with such issues. We're seeing this more and more with companies like Facebook, Google, Twitter, etc. What you have to understand is that you're not really a customer of theirs. You're their product - what they sell to advertisers. And how many companies can afford to offer phone support to all of the items that they sell?

About the only thing we can do to help is to help spread the word by printing this. Perhaps that will help reach the right person who can fix this mess, assuming you still want to use a service you have no control over and that offers this level of support.

Dear *2600*:

The Supreme Court's decision not to take up the ongoing debate on overbroad surveillance of American citizens at a sooner date should be reconsidered. This practice has a profound effect on the Fourth Amendment, which protects us from unreasonable search and seizure. "Third Party Doctrine" creates a loophole that can affect everyone's communications. Third Party Doctrine is basically when individuals voluntarily give information to others (such as corporations). A primary example would be telecommunications companies, where people give up personal data in exchange for services like Internet, email, or telephone without an expectation of privacy.

Free expression is a cornerstone of any free society and goes hand-in-hand with privacy because one without the other does not work properly.

Bill Miller

It seems every other day we're hearing of some other privacy violation that comes about when companies or institutions fail to safeguard the personal data they're entrusted with. We see hackers demonized and blamed every time, even when they clearly had nothing to do with it. By creating a scapegoat, the people responsible for security are able to escape responsibility for their inactions. It's not enough to protect our own data if the people we give it to don't take it seriously. We do have an expectation of privacy in such circumstances and we also have an expectation of responsibility when they screw up.

Dear *2600*:

Many thanks for including my article ("Take Your Work Home After Work") in the latest issue. I was very happy to read it!

One thing though - in the article I sent, the example code and the "execution command" both contained parameters inside triangle brackets. I can understand how these would have been stripped out via the html removal filters.

Many thanks again for publishing my article. You guys are ace!

Gerbil Byte

This was only an issue for Kindle subscribers and, once we were alerted to it, we were able to have the issue fixed and sent out again to replace the defective one. That's about as revisionist as we're prepared to get.

Dear *2600*:

I wrote *2600* while I was in jail. Did you ever get my letters or articles? I just had my case tossed after three years in jail. I would appreciate some sort of response.

Craig

The amount of mail we get is staggering so it's just not possible to send personal replies. We know it's especially hard for people who are imprisoned and we try as best we can to give them a voice in our pages when possible. We need to be clear that there's little we can do beyond that to fight people's cases. Over the years, we've had inmates send us all of their legal papers and daily updates in the hopes that we could somehow fix the system. We can't, much as we wish we could. But many have found relief by telling their stories through the letters pages, writing articles about hacking behind the walls, and taking out Marketplace ads to reach more people. Congrats on getting your case thrown out. That doesn't happen often.

Free Expression

Dear *2600*:

By reading this letter you have exposed your publication to a "poetry exploit."

It is a blatant attempt to earn myself the accolade of being printed in *2600* with the absolute minimum of effort. I hope you love it and feel compelled to send me a t-shirt!

The Hacker's Creed

I am a hacker
I have a hacker's mind
I cannot help but problem solve,
amongst the daily grind.

I am a hacker
I see through hacker's eyes
I find the underlying truths,
amongst assumptions and lies.

I am a hacker
I hone my hacker's skills
I take a thing, re-purpose it,
and bend it to my will.

I am a hacker
This is my hacker's creed
I search for understanding,
wherever it may lead.

StevieBohY

Not bad at all. Some of us feel this would work well musically as a black metal track, but that's just an opinion. However, while you succeeded in getting printed in our pages, this was sent to the letters department and we don't offer anything to writers other than the pride that comes with being published here. Articles are a different story, but then they're also significantly longer than letters. The letters section is the place to bring up any topic of interest, respond to other letters, tear apart or praise an article that was recently printed, or ramble on for no discernible purpose. And poetry can fit in there as well. In this age of 140 character communication, we hope to see more people take advantage of this forum of expression and immortality. Our address is letters@2600.com.

Dear *2600*:

Please post a link to your GPG key, with the fingerprint, on Twitter. I'm interested in submitting an article for publication... but would prefer a secure channel.

Joe

Our key is on our website in the submissions section. As we feared, we've already gotten several messages that somehow either mangled the key, used the wrong one, or are attempting to encrypt using an incompatible version of the encryption software. Please be certain you're familiar with the software and are using the proper key before using this for default communications. If you want to send us a test message first, we will respond if the message is decrypted successfully, although this requires manual intervention which may take some time, depending on our workload.

Dear *2600*:

I found the letter from Justin L. Marino in your Winter 2014-2015 edition disheartening to read. Here is a man who clearly wants to make good in his life, and educate himself and others, but is being stopped from doing so because the prison is scared its own computer security is not up to scratch.

Him being incarcerated got me thinking about the old cliches of smuggling tools in cakes into prisons. Perhaps the modern day version of this would be to have the text of *The Basics of Hacking and Penetration Testing* embedded in a modified copy of an innocuous book that would clear the prison censors.

With all of the self-publishing possibilities on the web these days, someone could easily scan portions of a proscribed book and another less controversial (in the eyes of the authorities) book, then merge them, and voila - modern-day saw-blade in a sponge cake.

This, of course, would no doubt be illegal, but perhaps budding authors out there might write a cyber security detective novel that gives full details about how the characters go about their business.

Rob

That's an ingenious and dangerous idea. The people in charge would have to read every page of every book to make sure it fit their specifications. These are practices that will need to be increasingly used outside of prisons as well since more and more of our lives come under scrutiny each year.

Dear *2600*:

The EFF has brought up something interesting about the TPP (Trans-Pacific Partnership).

This proposed regional regulatory and investment treaty poses massive threats to users in all sorts of ways. According to the EFF, "It will force other TPP signatories to accept the United States' excessive copyright terms of a minimum of life of the author plus 70 years, while locking the U.S. to the same lengths so it will be harder to shorten them in the future. It contains DRM anti-circumvention provisions that will make it a crime to tinker with, hack, re-sell, preserve, and otherwise control any number of digital files and devices that you own. The TPP will encourage ISPs to monitor and police their users, likely leading to more censorship measures such as the blockage and filtering of content online in the name of copyright enforcement."

Something for your analysis and enrichment.

Joethechemist

More like something to terrify and annoy us. There seems to be no shortage of evil legislation and ominous corporate agreements that wind up restricting access to a ridiculous level and ultimately controlling art and free expression to a stifling degree. We think everyone can come to an agreement on what constitutes criminal behavior and actual copyright infringement. The provisions being established with things like this are unhealthy and crippling. They ultimately will do more harm than good to the very industry that's promoting them. And we don't believe the actual creative talent responsible for all of the works in question benefits from any of this. When we all band together and oppose such draconian plans and agreements, then we will have an actual chance of producing something constructive and fair. Until then, we suggest frequently visiting eff.org and making plenty of donations so they can help fight this and all of the other ill-advised plans out there, as well as keep us updated on the newest threats.

Inquiries

Dear *2600*:

I found an interesting article that describes how payphones are being converted into Wi-Fi spots. If I send pictures of these hotspot/kiosks, will they be published in the magazine? Is a new form of phreaking in the works?

Joe

We can't guarantee anything, but we can say that the first real step towards getting published is always to send us something. Our payphone pages aren't always strictly payphones, so it's certainly possible this will find its way into a future issue. And, yes, a new form of phreaking is always in the works.

Dear *2600*:

I found a rather interesting news article on a virus called badBIOS, and I distinctly remember someone writing an article on a virus that kept rewriting their OS, even when they got a new laptop. I thought this could be the virus in question.

Josh, UK

Dear *2600*:

I've been reading your publication for ten years now. I bought my first copy when I was 13 while vacationing in Canada. I've loved every copy I've read. For that I thank you.

On to the important shit:

How can I give you the most money? Should I purchase a one year subscription every year or will the lifetime sub be more beneficial to you? What earns you more money? The subscriptions or clothing purchases? Any way for me to help beyond purchasing your publication?

Andrew

We've found this question being asked a lot recently, in the wake of what's been happening in the publishing world (declining print readership, bookstores going out of business, our getting massively screwed by distributors, etc.). It's extremely heartening to know that our readers have our back. But we never want to be soliciting funds unless we're giving something of value back. Buying something from us will always be beneficial. It's hard to say which is the best subscription-wise, as it depends on variables that change over time. If everyone bought a lifetime subscription, we'd feel great now, but 60 or 70 years down the road, when we were still obligated to send everyone a new issue every quarter, we might find ourselves struggling. Renewing every year offers consistency, but there's always the chance you could find yourself completely disinterested in our subject matter in only a couple of years. (It's happened at least once.) In short, we have no answer that works for everyone. One option that seems to be the best of both worlds is our electronic digital digest subscription, which provides digital access to all of our annual digests as they become available, doesn't involve extra resources to produce more copies, and which can be given as gifts to as many people as you desire and/or can afford. Thanks as always to our readers for thinking of us and for keeping all of this going.

Dear *2600*:

I was pleasantly surprised to see my photo and name on the back cover of the new issue of *2600*! Almost dropped the copy I was holding at the newsstand. Does this mean I won a subscription? If so, here's my address: [redacted]

Starting with the next issue of course, I'm buying a bunch of copies of this one to hand out to all of my family members for Christmas.

S

You should have received an email from us a few weeks after your material was printed. You can

then decide if you want a subscription or one of our t-shirts. Hopefully, all of that has already happened in your case.

Dear *2600*:

I'm a big fan of your magazine. I was wondering if you could recommend a good program to hide my IP. Thanks.

Chris

There are lots of proxy services and VPNs (Virtual Private Networks) available all over the net, some much better than others. A few you have to pay for and others are free. Anything we suggest here is likely to change over time, so the only way to really know what's good is to try them out. Please remember that such services can be used against you if they're not trustworthy or if they are compromised by hackers, governments, private eyes, etc.

Dear *2600*:

I found your address in an Amazon comment. I want to subscribe today for a yearly Kindle subscription. However, I was curious how many back issues I can access. 22:1 is my last printed copy.

Ratish

You have some catching up to do then. We've been on the Kindle since 27:3 and you can get every issue since then at the Kindle store. You can find out what else we have digitally by visiting the digital edition section at www.2600.com.

Dear *2600*:

I have a photo submission of a taxi cab in Boston bearing the number "1337." What email address should I use to submit it? (Assuming you are even interested in it - I know you usually look for "2600" but figured this was kind of cool.)

Nick

While cool photos of "2600" things are what most people send, we're really open to anything that relates to hackers or the net in a strange and "real world" way. So instances of words like "elite" and "hacker" would be right up our alley. The email address for any of these submissions is articles@2600.com. Please make sure your digital files are as good as possible and that you attach as much of a description as to what the images are and where they were seen.

Dear *2600*:

Hello, I have a few questions. Is there a deadline for an article to be published in the next issue? I was looking for a page on formatting, but didn't come across one. Is there one that I'm missing? Lastly, do you prefer articles in the body of an email as plain text or as an attachment?

Jon

As we're always working on one issue or another, there's no set deadline. If your article misses the hypothetical deadline for one issue, it will be considered for the next. Even if it makes our deadline, there might not be room for it in the next issue and sometimes even the one following it. Exceptions are always made for subject matter that's particularly timely or juicy. As for formatting, we prefer straight ASCII whenever possible, but we can read most formats that aren't too bizarre. It can't hurt to also send an ASCII version in case we have difficulty.

Dear *2600*:

Have you seen the remarks from British Prime Minister David Cameron on the need for new online data laws?

Xaus

Indeed we have, and we're both shocked and not surprised at the same time somehow. Leaders have a history of taking advantage of tragedy and terrorism and using such events as a means to push forward agendas they wanted all along. Remember, there is not a government on earth that doesn't want more of an ability to spy on its citizens. Sadly, we're seeing more of a trickle-down effect of this desire, ranging from local governments to parents. Everyone wants to be able to see what others are up to. But, to get back to what Cameron is proposing in the wake of the Charlie Hebdo massacre, nothing he's pushing would have been able to stop what happened. In most cases, surveillance of the masses does nothing but tie up law enforcement with a whole lot of data they have no business analyzing in the first place. However, identifying criminals, terrorists, and the like is still possible with decent detective work, the kind that comes from following leads based on things like actions and tips, not fishing expeditions. If you look at crimes that were prevented or criminals that were caught, you'll see that most times widespread surveillance had nothing to do with it. The words Cameron utters should be enough to make any thoughtful person see the threat: "do we want to allow a means of communication between people which we cannot read?" You can't make this stuff up.

Dear *2600*:

I have a Motorola Talkabout and was wondering if there were any phreaks I could do to it.

Josh

If by "phreaks" you mean increasing the power to increase the range, this is generally not seen as worth the effort as your battery life goes way down while your signal range isn't dramatically increased. If there's something else you're looking for, we'd need more specifics to be able to look into this.

Dear *2600*:

Is there a physical store I can visit? I am coming for a trip to New York City and would like to visit a store or something.

Adam

Assuming you mean a store of ours, you're out of luck. If we had to operate a physical store in New York City (or anywhere else, for that matter), we wouldn't last very long, mentally, physically, or financially. We're afraid it doesn't get any better than our online store or occasional appearances on tables at various hacker conferences. We hope you find other stores to visit in New York City - there are quite a few.

Dear *2600*:

This probably will sound like a really dumb question, but how do I make use of the different code you have posted on your website in the "code" section? I am just learning Python and am obviously a noob when it comes to coding, but would greatly appreciate the help! Also, what program/programs could I use to utilize the source code written for i-devices? Any help would be great!

P.S. Thanks for sticking it to Sony! Screw those guys!

Brian

Re: Sony, we just felt it was time to remind them what it means to take a stand and not cave in to threats. We know they're usually on the other side of that equation.

Concerning our "code" section, it's different for every article. Sometimes people include code snippets in their articles and other times it's entire programs. Depending on what they're written in, you will need to use different methods to get them to work. The more you learn about programming, the easier it will get to decipher and apply. Concerning doing more with your i-devices, we suggest reading the Wikipedia page on "iOS jailbreaking" as it explains a lot of this in great detail. We can't stress enough the importance of knowing what you're doing before embarking on this particular journey.

Dear *2600*:

Does *2600* have a position on climate change? Toronto350.org is one of many groups working to build a safer future by controlling climate change. We might be able to write an interesting article about our experiences so far. Let me know if that sounds at all interesting,

Milan

Our position is simple. Science tells the story. If we pay attention to the data presented, the facts are inescapable. Those who believe science has some sort of political agenda basically have a medieval mindset and need to be bypassed if we want to actually accomplish anything. We trust that answers your question. As for an article, just remember to think like a hacker when writing it. There's no subject where that mentality can't be used to come up with solutions nobody ever considered before.

Dear *2600*:

If I order a subscription and select "Winter" as first issue, would I get the 2014-15 Winter issue? Or not get my first issue until Winter of 2015-16?

Brandon

That would really be nasty of us to make you wait an entire year. We have options at store.2600.com to begin a subscription with either the current issue or the next one. This way, if you buy an issue at a store and then subscribe, you won't get two copies of the same issue.

Dear *2600*:

I am brainless and am guided by saints/hackers/radio people, so I have made no contribution in life whatsoever. I am also lacking in education in comparison to the status quo. I also am not a hacker/cracker/phreaker/scientist/educator/lawyer/doctor but I am quite lazy. Here is my question: Is there a website that I can go to that will give me access to free satellite television on the computer that I use for Internet access at the library. My time limit is 90 minutes while online. I do not have Internet access while at home and my only freedom (haha) is while I'm here at this library in Texas. Forgive me for the broken English. I am not a smart person like all of the people that contribute to this periodical.

stupedestrian

The first thing you need to do is stop saying such nasty things about yourself. If you're capable of asking a question, then you're capable of learning and making things better.

Assuming you have access to a pair of headphones while in the library (so you don't annoy everyone around you), this shouldn't be too difficult. But you may have a problem finding the exact channel you want if they don't have a live stream on their website. You can look at sites like streema.com to see the kinds of things that are available. Be prepared for spotty connections and unpredictable content. It's all part of the fun. If any of our readers have additional suggestions, please send them in.

Dear *2600*:

An interesting thing happened to me today that I need clarified. Only the people at *2600* are qualified to help me resolve this issue and so I am writing you for your help. I called the number 1-202-456-1444 and got the recording "You are about to activate the government management scenario. Please enter the access PIN followed by the pound sign." I cannot figure out what this is or what it means. Please ease my worried mind and explain this. Any and all insight that you can provide will be much appreciated.

Brainwaste

We've never been able to get that recording despite the many times we tried calling (no doubt, we've now generated another government

file on us). We can say that this phone number is somewhere within the White House and, according to our archives, was once listed as belonging to Richard Nixon. Now it seems to go to silence, which seems appropriate.

Tribute

Dear *2600*:

I don't know if you ever carry obits, but in case you'd consider it, I've written a piece about Steve Gold who passed recently. Steve was a good friend of mine, but my reason for sending this is his significance to the hacker community.

In the U.K. of the mid 1980s, no one really knew if hacking was illegal. Steve Gold helped clarify that situation - by being prosecuted by one of the largest organizations in the country. Thirty years later, on January 12, 2015, Steve died peacefully in hospital. But he left behind a legacy of great significance to the hacker community. An ex-nurse who became a senior auditor and fraud investigator for the National Health Service (NHS), Steve had hacking in his blood. Three decades later, he could still recall in intricate detail his phreaking adventures on the nation's phone systems. He was part of an early 1980s scene that encompassed all that is best of the hacking mentality - an unquenchable curiosity and a mischievous disregard for petty rules.

In the mid 1980s at a computer show, his friend Robert shoulder-surfed an engineer from British Telecom (BT) logging in to the Prestel system. This was a Viewdata service that carried news, weather, share prices, and much more. In 1983, Prestel started to carry a new service called Micronet 800 for home computer enthusiasts. Steve would become one of three people who ran a section of Micronet known as Micromouse (and until his death was still known by the nickname Skweek by many friends). Micronet also offered a primitive form of email. In those days, a Prestel login consisted of a nine-digit ID (usually the customer's phone number) and a four-digit password. The engineer's credentials were 222222222 and 1234. With that information, Robert and Steve began to explore Prestel with super-user privileges.

There is so much damage they could have done - such as changing share prices or taking the system offline. But what they became notorious for was reading Prince Phillip's messages. BT tapped their phones and eventually pounced. In 1985, the two men were arrested and charged. But here was the difficulty. What was the offense? They could have been charged under the Telecommunications Act, which makes it illegal to incur charges on anyone else's account without their permission. But BT wanted to set a precedent. It needed to make it clearly illegal to exploit another person's credentials even if the service is free and no charges are incurred. So they went with a charge of forgery. The argument went that the login process essentially created, for a moment, a forged "instrument" - an authentication setting in the computer's memory.

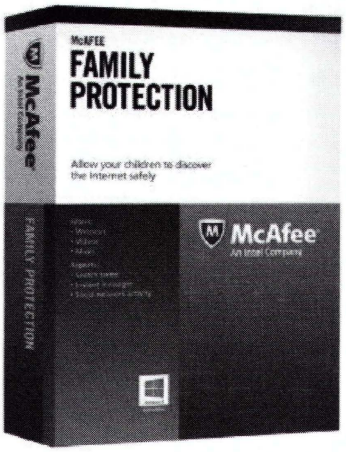
BT won and both Robert and Steve faced stiff fines. But they appealed, and won. BT wasn't content to give up there - it needed this conviction, and so the case went to the House of Lords where the acquittal was upheld. According to a private source, BT had spent something in the region of a million pounds prosecuting the case and was left no better off. Steve and Robert were vindicated and the authorities in the U.K. were left with no doubt that legislation was needed to deal with this new phenomenon. That legislation came with the Computer Misuse Act 1990.

Steve turned his knowledge to good use. He became a successful, popular, and prolific IT journalist, covering every aspect of the field, but always with a special love for security. He was a frequent speaker at security conferences, often chairing panel sessions, and also gave many lectures alongside the police officer who arrested him. Late in life, he took a degree in psychology and lectured on the psychology of hacking at a couple of universities. His students were often members of the intelligence services and police force cybercrime units. I worked with him on many magazines and projects over the course of nearly 30 years. And, a few years ago when I took on the editorship of two specialist journals - *Network Security* and *Computer Fraud & Security* - he was the first person I turned to for insightful and thoroughly researched contributions.

When he died from complications following heart surgery, Steve was two days short of his 59th birthday. This has robbed the infosec community of Steve's wealth of knowledge and experience - but most of all we have lost a kind, loyal, and generous man who embodied all that is best in the hacker world.

Steve Mansfield-Devine

Thanks for this most deserving tribute (so much more than what usually defines an obituary). The Prestel story is one from our early days as well and there's a special connection between everyone who was involved in the various exploits of that time, one that continues to this day and has included many from younger generations who see the importance of this history. There are so many stories in our world that deserve telling. We believe you've touched an entire community with this one.



McAfee Family Protection - Epic Fail!

by Brian Van Stedum
brianvanstedum@gmail.com

Last spring I took a security fundamentals class while pursuing my degree as a network specialist. This class was the most challenging, yet rewarding, of all of the classes that I have ever taken. The final project for the class was to find a security software suite to analyze and ultimately to circumvent its security. My instructor recommended choosing a suite that focused on parental controls, and I chose McAfee Family Protection. McAfee Family Protection is designed to give parents the ability to control and monitor how their children use the Internet in order to prevent them from accessing potentially harmful information. I chose it simply because of the name: McAfee. I wanted to be challenged and I figured, "Hey, the DoD recommends McAfee's antivirus software. They must be pretty good, right?" It didn't take me long to realize just how wrong I was.

Just a little about the testing process: all tests were conducted using Windows 7 and most were conducted using Windows administrator account access. I performed the tests while using administrator access since circumventing Windows user account control security can be easily done by using a boot disk containing Ophcrack or NTPW. The following analysis is the product of an in-depth audit conducted to discover any and all methods to circumvent the security of MFP. This was not a test of its effectiveness; I did not care if it let a couple of porn sites by its filters. The goal was to find any way to bypass its individual security features entirely. The analysis is broken down by each successful circumvention.

The majority of MFP's configuration settings, including authentication settings, are stored remotely on McAfee's own servers. The absence of locally stored configuration files initially made circumventing its security a little more challenging. However, after analyzing the software further, I discovered many other methods to successfully bypass the software's security. MFP also did a fairly decent job of protecting its own locally stored files from alteration and removal. However, it did not provide any type of protection for the Windows environment, which allowed me to perform tests and alter the system in order to bypass MFP's security.

MFP creates a usage log for all users that can send daily reports to the account administrator. I discovered that the log is stored locally and was only sent to the online servers once per day. Upon inspection of these log files, I determined that the file itself was not user readable and was also protected from alteration and deletion. However, I was able to change the file's attributes, and by setting it to read only, I was able to prevent any future Internet usage logging for that day.

MFP's Program Blocking feature blocks programs from accessing Internet resources. An administrator can specify which programs to block based on a suggested set of programs or specify any other program to block. I was able to bypass this feature by simply changing the name of the executable file for the program that had been blocked.

MFP's website blocking feature allows the administrator to block certain websites that the content filter would not flag as harmful. This feature is easily bypassed by adding an entry in the Windows "hosts" file that points to the IP address of the blocked website, but uses a domain name from a site that is not blocked.

Although MFP is pretty decent at content filtering and protecting its own files, I was

able to easily bypass the entire security suite by booting into Windows "safe mode with networking." By logging into safe mode, I had unrestricted access to the Internet and was also able to circumvent McAfee's protection of its files.

The services that MFP uses cannot be disabled, stopped, or paused even while using the Windows administrator account. However, by using Windows safe mode, a user can change which services load at Windows start. By using Windows safe mode and registry editor, I was able to change the startup mode of the three main processes used by MFP by changing the DWORD "start" values from 2 to 4. Once I rebooted back into normal Windows mode, McAfee Family Protection was completely disabled and I had unfiltered Internet access.

MFP's literature boasts about how secure its uninstallation process is; it uses a unique uninstall key, which is only good for 24 hours, and requires an uninstall program that can only be used by the MFP administrator. As secure as they think this process is, it can be easily bypassed by using Windows' built-in system restore function. A Windows administrator can select a restore point prior to when MFP was installed to effectively remove it from the system.

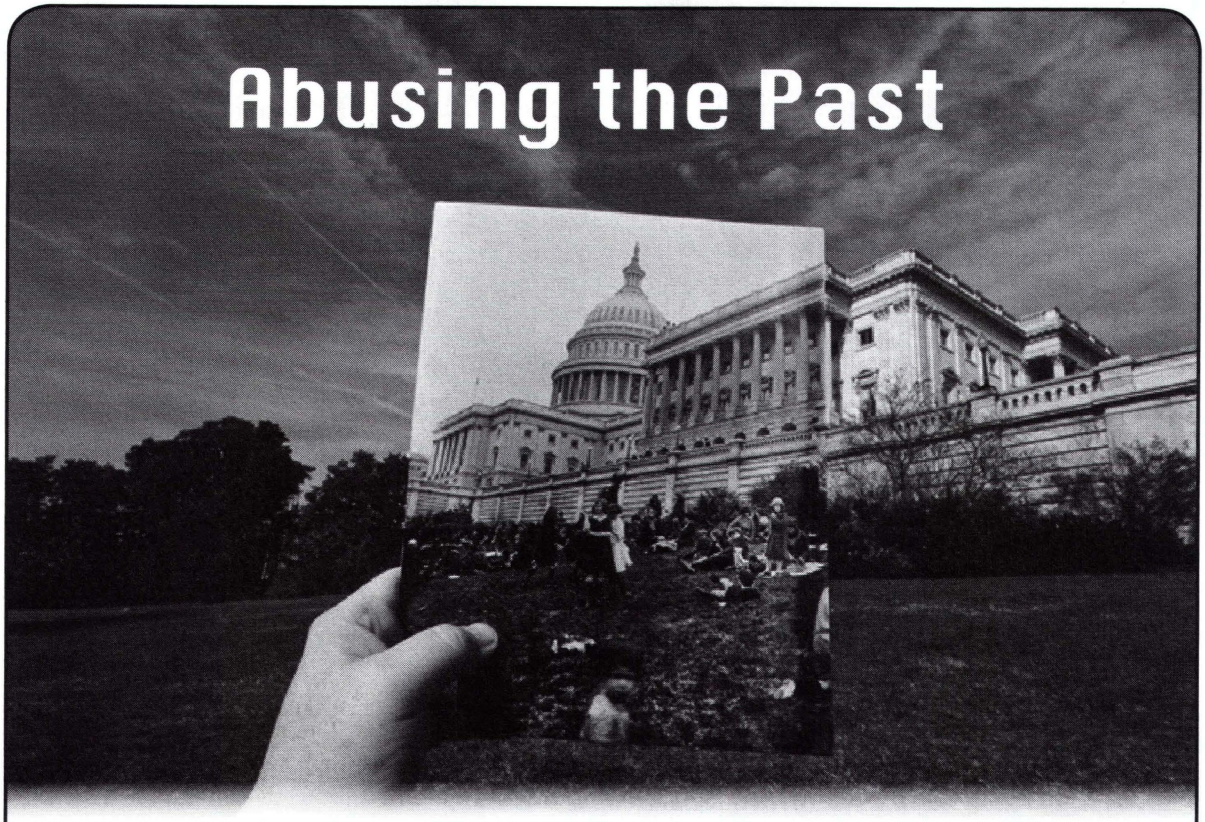
After exploring the many files MFP installed on my test system, I observed that it installed all the language conversion files on the system (not just the version I chose). After decompiling a few of these DLL files, I discovered that the file mfploc_en.dll also contained the many keywords that were used by the safe search feature. I found that altering or deleting this file was nearly impossible, as it was protected by McAfee. By utilizing the Windows safe mode loophole that I mentioned earlier, I was able to remove the mfploc_en.dll file and rename mfploc_ko.dll to mfploc_en.dll. By doing this, I was able to change the language from English to Korean. Since it was now searching for Korean words rather than English words, I was able to search for any term I wanted to without being blocked.

Upon initial inspection, it appeared that MFP's greatest strength was that it saved the vast majority of its configuration files to McAfee's servers, rather than on the local machine. However, after examining the changes it made to the hard disk, I discovered that MFP sends most of its initial configuration changes via crafted html files that, once sent, are saved in the "Temporary Internet Files" of a Windows 7 system. After reviewing the saved configuration html files, I found one that used the administrator's username and password as an argument for the file, which it displayed in clear text.

After reviewing the saved html files that were sent by MFP, I discovered one named "239" that contained some local system information as an argument. After re-executing this file (by simply double clicking on it), a web browser opened with an administrator login prompt that was meant to be used to associate the local installation with the online McAfee user account. By going to McAfee's website and signing up to obtain a trial of MFP, I was able to create a username and password that would become an administrator account on any new installation of McAfee Home Protection. With this new account in hand, I entered the account information in the prompt that opened by executing the "239" file, and then associated the local installation of MFP with the administrator account that I had just created. Since this was a new account, it would be impossible for the administrator of the original account to discover this new admin user. Not only does this new administrator hijack the local installation of MFP, but it still permits the users of the original account to login on the same machine.

Throughout this analysis, I was continually shocked at just how easy it was to bypass McAfee Family Protection's security features. Additionally, I was able to bypass MFP's security by utilizing methods such as an online VPN service, a remote desktop connection, a live OS on bootable media, among others. I performed this analysis over the span of two weeks while having to study for other final exams, work a full time job, plus attend to my everyday family obligations. Yet I was able to completely circumvent MFP's security features. For a motivated teenager with nothing but time, bypassing MFP's security features would be a walk in the park. After completing this analysis, I believe that McAfee Family Protection is ultimately useless due to the fact that a child with an average knowledge of computers could easily bypass its security.

Abusing the Past



by Buanzo

Disclaimer: If you do evil shit with this information, I hope something really bad happens to you. Information is free, but people are human.

In this day and age, there are mass scanning tools and several easy-to-query databases that make it a simple thing to find sites with vulnerabilities. Hackers and other agents with all hat-colors use them every day to do their jobs. I will present to you today a very simple technique that will, when certain special circumstances are met, allow you to scan the past for vulnerabilities.

When we want to have a website, we obtain a [sub]domain name, point it to some web hosting server's IP, and configure it to serve that website. We also get DNS service somehow. I am sure you've done this before, so I'll skip those details. So now, www.example.com is running on server A.

Yay, we've got a website! By the way, it is Joomla or some other CMS like wordpress, etc.

The days/months/years pass, and we find ourselves needing to move the website to another server, for whatever reason (luckily, because we have so many visits, the old server can't handle them). The new website is configured on the new server, the DNS is updated, and voila, visits now arrive at the new server.

Nice.

But....

If we go to `netcraft.com` and check some domain name using their tools, we *might* find the hosting history of a website. Yes, www.example.com used to run on server A, then server B, now server C! And, wow, that's weird, the old servers are still up and running.

So, www.example.com *might* still be configured in one of those servers. You know how hosting companies [don't] do their homework sometimes!

So an attacker could fire up a scanner, and by any means available, target www.example.com through the older IP addresses, and scan our old *website[s],* which, of course, we no longer keep updated (maybe not even the server, for that matter...). And you know what outdated usually means: holes. Lots of them.

And holes lead to lots of things: remote code execution, data exfiltration, resource control.

An Nmap NSE script could be written to scan some domain name's hosting history, and, essentially, abuse the past.

Go. Check your hosting history. Don't say I did not warn you.

Hacking the HandLink Gateway

by secuid0

Many cafes, restaurants, pubs, and other shops offer to their customers Internet access through Wi-Fi as they know that it's pivotal for drawing in customers and securing their repeat business. Usually, all customers have to do is buy a cup of coffee and enjoy free Internet for x minutes. In some other cases though, shops are preferring to get some revenue out of this service, which means customers have to purchase a Wi-Fi voucher directly at the counter.

One of the most common low cost deployed solutions which handles the authentication, authorization, and accounting for the Internet access is the HandLink WG-500P. This is a small wireless subscriber gateway. It's dead easy for non-tech-savvy staff to operate it; the store representatives with the press of a button can issue a voucher which is printed through the built-in thermal printer.

In order for the customers to use the voucher, first they will have to connect to the cafe_wifi. The captive portal (pointing at `http://1.1.1.1`, `http://192.168.1.1`, or `http://192.168.88.251`, etc.) will prompt them to enter a valid username and password into the login form. If the combination is correct, then access is granted.

Now let's imagine the below scenario:

1. We are at a nearby location where cafe_wifi has coverage.
2. We are neither hungry nor thirsty.
3. We need access to the Internet to download an ISO or the latest fappening leak.
4. We may or may not have left our wallet and credit card at home.
5. The shop is using these nifty WG-500P machines.

One thing we can do is point our browser at `http://10.59.1.1/` (this is the internal LAN IP address of HandLink WG-500P) and try the following username/password:

1. `admin/admin`
2. `supervisor/supervisor`
3. `account/account`
4. `super/super`

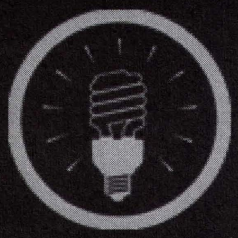
Chances are you will find combinations #1 and #2 invalid, but not #3 and #4. Once you login, then issue the following POST request (to create the request, you may use burp, OWASP ZAP, and/or if Firefox is your favorite browser you may use the Hackbar addon - it's pretty simple):

```
http://10.59.1.1/webAccount
➥Generator.cgi
POST data: "button=0&webAccount
➥GeneratorHandler="
```

On the spot, a voucher will be generated for you and will be displayed on your screen. Use the newly created login at `http://192.168.1.1` and voila, profit. Although the whole approach may or may not work and cannot be considered as a fancy hack, it's worth trying.

Happy surfing.

EFFecting Digital Freedom



by Parker Higgins

The meeting room of the Federal Communication Commission is an odd place to see a victory for the kinds of digital civil liberties that hackers hold dear. But there it was in February, that after being targeted by over a year of nonstop activism campaigns, FCC Chairman Tom Wheeler led a vote in favor of strong net neutrality rules and delivered remarks on the importance of free speech online that sounded more like John Perry Barlow's "Declaration of Independence of Cyberspace" than something from a former cable lobbyist and current top regulator. Wheeler insisted that these new rules are not, as critics charge, an effort to regulate the Internet; to the contrary, he said, net neutrality is no more a plan to regulate the Internet than the First Amendment is a plan to regulate speech.

How does that hold up? Here's what we know: the FCC approved a plan to place Internet service providers under a different (and stricter) title of the Communications Act, and to prohibit those services from engaging in site blocking, throttling, or paid prioritization. That last point prohibits the kind of "fast lanes" that had been tossed around in earlier proposals, and which would have led, naturally, to "slow lanes" as well - at odds with the basic principle of net neutrality. Reasonable minds can disagree about how close regulators should get to the Internet in the first place, or how effective these rules will be, but these, at least, are noble goals.

There are, however, still some critically important things we don't know. First things first: the entire process could use a lot more transparency. For example, it seems anathema to the spirit of Internet policy, but even as the FCC vote took place, the actual text of the new rules was not available for the public to read. There's been a lot of finger-pointing between commissioners about why that's the case, but sadly, it's the way things go with the FCC. That lack of transparency is one reason EFF has been skeptical of the agency for years.

It's especially important in this case to see the actual language, because the FCC may have used the kinds of weasel words that could allow bad behavior from ISPs, or leave the rules themselves open to legal challenge. For example, the prohibition on site blocking extends only to "legal content." We'll have to watch carefully to make sure that language isn't used to draft ISPs into fast-and-loose vigilante copyright enforcement, for example. Similarly, plenty of pundits expect one or more of the ISPs to sue to block the rules; if that happens, ambiguity in the language could weaken the FCC's case.

Net neutrality is an important goal, but considering these factors it's a bit premature to say for sure we've gotten much more than a mixed bag. But even if it doesn't close the book on the Internet's efforts to achieve net neutrality, it will certainly remain an interesting chapter. For one thing, this is a story where conven-

tional wisdom proved completely wrong. As of January 2014, in the wake of FCC's last major courtroom loss to Verizon, it was universally held that net neutrality was toast and the agency would never find the political will to undertake the reclassification that could save it. When EFF joined a large and incredibly diverse coalition of activists to push for that outcome, it was a moonshot, but 13 months later the coalition won.

Taking a step back: that vote is the latest in a string of apparent victories for computer users over forces that have historically been able to shape laws, regulations, and even market conditions. In just the past several years there were also, of course, the twin victories over the Stop Online Piracy Act and the Anti-Counterfeiting Trade Agreement in early 2012, and the massive push for more secure and private online services in the wake of the Snowden revelations in June 2013.

Each of these developments were influenced by countless factors, but they have some important elements in common. Substantively, each represents a victory for hacker core principles - freedom of speech, freedom of privacy, and the freedom to build new things, or play with old ones, without getting permission first. Tactically, though, the overlap is even more pronounced. In every case, people harnessed tech to amplify public voices in ways that politicians and executives didn't know to expect. It's been more than just moving traditional activism online - there's been the kind of creative and playful problem-solving that we've always known is part and parcel of the hacker community.

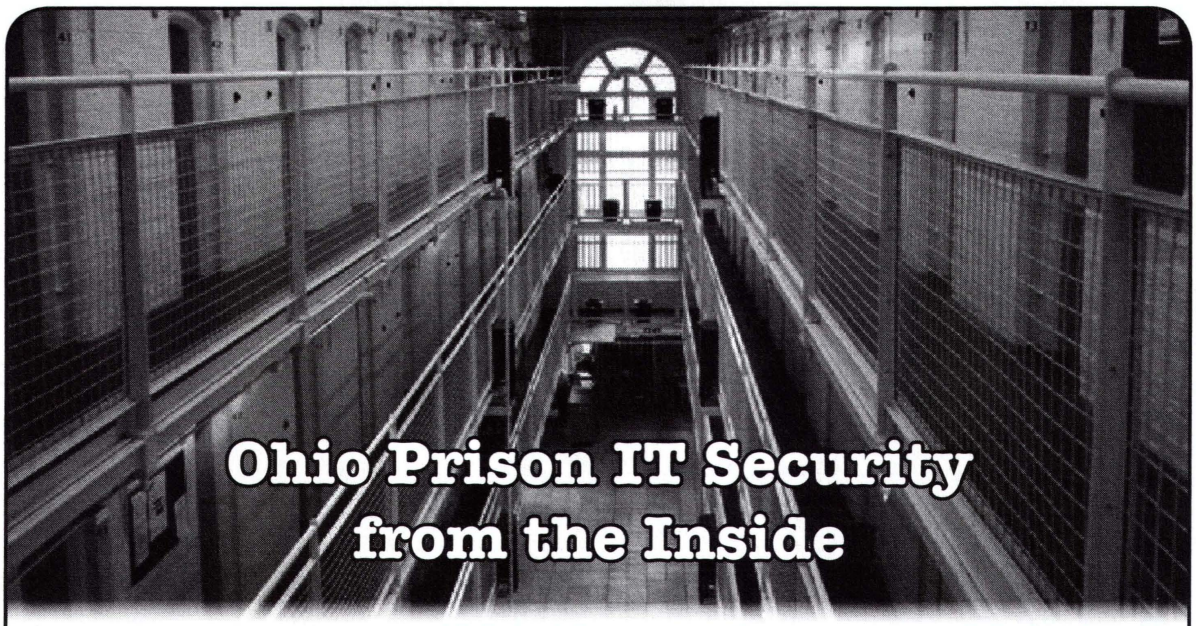
Many hackers express a desire to keep out of politics. Tech wouldn't go to government. But since networks have pushed into everybody's lives, government came to tech. For at least the past several years, it hasn't been an option to just ignore what the politicians are doing. As EFF continues work on these issues, major battles loom: legislative reform of the NSA and other intelligence agencies' surveillance practices, the eradication of DRM software and the laws that prop it up, and a sorely-needed rewrite of computer crime laws like the Computer Fraud and Abuse Act, to name a few. If we're going to win - and we must - we'll need inspiration and help from the hacker community.

SUPPORT THE EFF! Your donations make it possible to challenge the evil legislation and freedom restrictions we constantly face.
Details are at https://supporters.eff.org/donate.

Lifetime PDFs - Volume 6

Come and join the lifetime digital digest club. You'll get all of our existing digests, plus a newly archived one every quarter, along with a brand new digest once a year for as long as you or we are around. $260 gets it all. Latest releases: Volume 30 from 2013 and Volume 6 from 1989.

Visit store.2600.com and click on PDF Downloads.



Ohio Prison IT Security from the Inside

by 5MEODMT6APB

Prison is not a nice place. It is an environment suited for predators, fighters, and schemers. Intellectual prowess only gets one so far in here. The ability to observe and adapt is one's best tool.

I have spent a lot of time observing what comes naturally to me: IT security. To put it mildly, the Ohio Department of Rehabilitation and Corrections has a lot of opportunities for improvement.

The most apparent failure is the culture of the IT department throughout the state. Due to budget constraints, there are only a handful of employees to manage the IT infrastructure of 20-something prisons. The management theory appears to be reactive instead of proactive due to limited resources. The dedicated on site IT staffer is poorly trained and not security conscious.

One would think in a prison setting that security would be a prominent theme when deploying new assets, but it seems to be an afterthought.

Staff and most inmate computers are physically segmented on their own networks. Most inmate-used computers are for educational purposes of one sort or another. In most cases, they are on their own domained environment and authenticate a general purpose account to a DC. Group Policy is employed to limit local access and prevent configuration changes. In addition to GPO, a software program named Fortres is used to secure the desktop. Two major implementation flaws exist in this setup: 1) Fortres can be defeated by opening the config files in edit.com and corrupting them. Much more simple: 2) the local administrator account is left enabled with a blank password. In fact, the XP image used by ODRC on inmate computers contains a blank password for the local admin account. No real security threat exists by having open access to a segmented network computer, but it demonstrates the culture.

Interestingly, the law library computers run a live Debian distro that has been customized by LexisNexis for access to their web-based law research system. These computers are connected to a VLAN which ultimately touches the Internet via an Internet-facing proxy server that is set to "Deny All Bidirectionally Except". It allows traffic to LexisNexis and to a secured section of ohiomeansjobs.com, both of which serve compartmentalized resources. Any attempts to influence redirects or otherwise access resources not permitted by the proxy fail at the network level. ICMP traffic is also denied to both internal and external resources. Overall, the law library and job assistance computers are secure and only subject to local vandalism.

ODRC has recently contracted with JPay to install terminals in the recreation and housing areas of the prison. These terminals allow for civilians to correspond with inmates via jpay.com. The implementation of these terminals, however, is patently insane in this hacker's opinion for the following reason: they are connected to the operational staff network. JPay and ODRC apparently bank their security for these terminals on software called SiteKiosk, which runs on top of the Windows 7 desktop, but under jmailinmate.exe, which is the JPay software. The SiteKiosk software works at a low

level to prevent the jmailinmate.exe program from losing focus or being closed, among other tasks like managing updates and desktop security. If jmailinmate.exe hangs, the Windows dialog box appears and prompts the user to force close or wait. If a force close is executed, the terminal is effectively stuck and secured at a JPay splash screen. Pressing escape at this splash screen brings up a "service personnel administration login" which is nothing more than a SiteKiosk password prompt. To my knowledge, this password has not been compromised. It won't be long, however, as one is offered as many attempts as they like.

Finally, the most glaring flaw is that during reboot, which occurs frequently because the terminals are constantly at issue and are restarted either remotely or by the SiteKiosk software, there is an approximately three minute time frame where Windows has booted but SiteKiosk is loading and starting services. The long time frame is likely due to disk fragmentation, huge log files, and poor configuration. During those three minutes, one can bring up the sticky keys context menu. From there, drilling up to the Control Panel is a two-click task. Clicking on Network Neighborhood populates with every single staff operational computer. From there, proper permissions and resource security are the only things stopping a major incident. This particular hacker was, as we say in prison, STD - scared to death to continue on any further. If previous performance is any judge, resource security is likely haphazard and pieceworked.

Lastly, one can click on external links in the Windows hung application dialog box, which returns a customized SiteKiosk-branded DNS error. DNS appears to be handled by a hosts file or through a proxy.

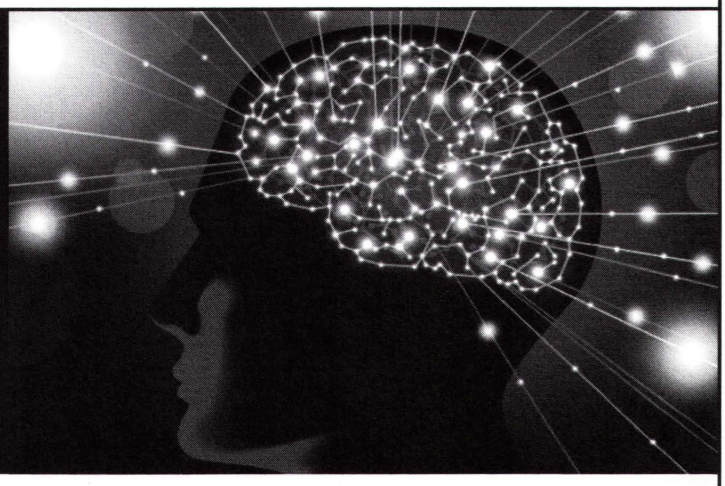
Cell phones are a major contraband issue in the Ohio prison system. The poor security of inmate used desktops only eases unmonitored communication with the outside through the use of USB cellular modems. No electronic countermeasures such as hidden femtocells or jammers have been observed to thwart smuggled cellular devices.

Overall, security is a joke inside the Ohio prison system as demonstrated recently by an inmate placing a ladder on the fence of a maximum security prison in Mansfield and climbing over. There is a massive drug problem fueled by enormous profits for both inmates and guards and a culture of laziness and passing the buck which prevails.

Perhaps this article will spur competency and a realization that inmates are not as stupid as they may appear at first glance.

Shouts to onestein, Aganthorp, Shrub Art, and flow. Late.

Hacking For Knowledge



by Jerry

Defined by Wikipedia, "Hacking" may refer to:

- Computer hacking, including the following types of activity:
  - ◊ Hacker (programmer subculture), activity within the computer programmer subculture
  - ◊ Hacker (computer security), to access computer networks, legally or otherwise
  - ◊ Computer crime
- Phone hacking, the practice of intercepting telephone calls or voicemail messages without the consent of the phone's owner
- Illegal taxicab operation
- Pleasure riding, horseback riding for

purely recreational purposes
- Shin-kicking, an English martial art
- The act of stealing jokes
- Hacking, an area within Hietzing, a municipal district of Vienna, Austria
- Roof and tunnel hacking, a type of urban exploration

"Bollocks," I say. My desire to understand how things worked, my unending curiosity, combined with insufficient funds, required me to repurpose cast-off computer hardware for experimental uses.

Or perhaps that should read "Mental" uses.

A Brief History

Early on (1960s), I built a light meter for my photo darkroom, obtaining a parts list from an obscure photo magazine. Success encouraged me to pursue additional adventures in creating needed hardware without sufficient funds.

I needed a set of transmission "jigs" for a VW Type 1 vehicle. It was built with a few scraps of angle iron and some effort. Of course, the key to this success was the age-old expression, RTFM. I scoured the VW manual for dimensions and such, including proper assembly procedures.

The Sinclair ZX81 at only $99.95 was the first computer for under $100. The ZX81 had the same microprocessor and ran at the same speed as the earlier ZX80, but it had a better BASIC programming language and was cheaper to produce. I had purchased a Sinclair ZX81 just before the IBM PC came out. I had my son programming in BASIC at the age of eight. Way to go, dad.

Fast forward into the 1980s, and the IBM PC was all the rage. I had to have one. However, the buy-in was way above my pay grade.

Enter the IBM clone. Eureka! I had a computer.

After a while, building your own computer was all the rage. And build I did.

I'm a "hands on" guy and that's just how I learn. And learn I did, creating the first PC network for L.A. County in the late 1980s, starting the move from "dumb terminals" and mainframe to "client server" with PCs.

Fast forward again (1998). I selected FreeBSD UNIX for my students to study. As a college program director (I retired from L.A. County), I chose UNIX over Linux simply to give the students a wider range of study. However, Linux was used also.

These days, I find the surplus computer market to be loaded with hacking/learning opportunities including the latest versions of Linux. I utilize at least one desktop and one laptop in my lab to explore the various Linux and UNIX distros available. Purchased through the surplus computer distribution channel, these low prices are affordable for all.

The Phoenix Project

Regular readers here will remember stories about data recovery from surplus hard drives. All true and, even better, complete computers with all hardware and software intact. Enter my latest surplus computer purchase, a "Supermicro" with Intel motherboard, 19 inch rack mount server, (one U) running windows Server 2003 with C.O.A.. The good news is: $50 out the door. The bad news: the administrator account had a password. Well, not too bad, as I had collected a Linux boot disk that ran a script allowing me to delete the administrator password in any Windows version. This operation takes about five minutes or less. Google will steer you in the correct direction for your own boot disk. Here's a tip: If you are working in IT support, don't let your customer know how easy it is to delete passwords. (It's bad for business.)

Here comes the knowledge. It turns out the server was a fully configured FTP server for a high end electronics lab (name redacted). Full virtual setups including server instances, NICs, and services. VMware headed the list of software included.

Here comes the "Best Practices." Included I found all setup software and passwords in plain text format. Score! Without a BIOS password, the system administrator password was simply reset, allowing full access.

The server didn't directly connect to the Internet due to the proxy settings configured for the lab domain controller server. Once that was corrected, and enabling DHCP, all systems were go.

(The server was purchased from `http://www.siliconsalvage.com/`. Fine supplier of all types of electronic surplus, and they also rent movie studio electronic props.)

I'm still exploring the wide selection of software available on this beautiful rack server. I encourage you to follow the yellow brick road of discovery as I did. Next up for this server, a full Linux install with Cloud infrastructure. Never give up, never surrender. Knowledge is free for the taking - grab it.

Linux Containers for Event Training

by Jon Schipp
jonschipp.com

Goal: To enable organizers and presenters of information security conferences, Linux user groups, and *2600* meetings to quickly prepare and serve training environments that teach and demonstrate Linux-based software to participants. By reducing the administrative overhead and the barrier to participation, we can improve the overall quality of training at events.

It can take hours to package and distribute a virtual machine with the necessary tools for training, and now it can be done in minutes including deployment using Docker containers.

Background

Software demonstration and hands-on training improve the experience of attendees during community events by not only sharing information, but allowing it to be practiced, which yields greater retention, understanding, participation, and fun. However, the logistics come at a high cost for both the user and the administrator. Virtual machine, or virtual appliance, based training tends to be the most common form, allowing a large number of participants to follow an instructor through an isolated environment, each running on their own computers. Using virtual appliances, while a workable solution, is not ideal due to the amount of time involved in their preparation, distribution, and configuration. Shared machine training is another form where users are given accounts to a UNIX-like system which they can remotely access.

The concerning problems of both methods can be summarized in a brief list:

a) Too much time is spent distributing, downloading, or copying virtual appliances

1) Conference networks are slow and VM files are big

b) Technical difficulties can and often will occur which end up putting some students behind others

1) Hypervisor image compatibility e.g. Virtualbox, VMware, etc.

2) VM bus and network configuration

c) Account management is repetitive and time consuming on shared systems

d) Changes are not easy in virtual appliances

1) Insertion of wrong exercises, versions, mistakes, etc.. How is this handled?

Linux-based Containers

Linux kernel 3.8 introduced the building block for containers, a form of lightweight process virtualization, or operation system level virtualization[1]. The two building blocks are namespaces and cgroups. Namespaces provide resource isolation, effectively making a system resource believe it's a part of a global resource through abstraction. There are six namespaces at present and they include: pid, net, mnt, ipc, uts, and user. pid, for example, allows processes applied to a namespace to be isolated from processes in other namespaces. Control groups, or cgroups, is the mechanism to which constraints can be applied to resources such as limiting the CPU and RAM usage to processes in a particular cgroup. This type of virtualization is done at a higher level, as opposed to the lower level hardware virtualization used in virtual machine technology. A benefit is that containers do not impose as large a cost by sharing the same kernel. Container startup time can be around 100ms, reaches near bare-metal performance, and outperforms KVM virtual machines in a wide array of applications from disk to memory[2]. With this comes greater density, where hundreds or thousands of containers can run on a single system. In addition, from the general user's perspective, having a shell inside a container or virtual machine is indistinguishable.

There are a number of userspace container runtime implementations, including lxc, Google's lmctfy, systemd-nspawn, Docker, and the newly announced Rocket runtime. Docker, a container runtime and deployment platform, is currently the most widely used, and for this reason my choice as the technology behind ISLET.

Isolated, Scalable, and Lightweight Environment for Training

ISLET is a solution for teaching Linux-based software with minimal participation effort by using Linux containers, and satisfactorily addresses each item in the aforementioned list of problems. It's a wrapper around Docker, SQLite, and a few other tools that in effect

reduces preparation and deployment of training environments to a simple three step process, enabling you to have ready to go training environments in minutes rather than hours. Account management is automated and handled internally by ISLET and is separate from the host, which allows users to resume their work (by reattaching to their container) should training events span multiple days.

ISLET is intended to be run as a server which students can remotely access. One single host account is required for ISLET which can be shared with all participants, its shell is set as islet_shell which handles everything after the initial authentication to the host. The participation barrier is set very low, and students only need an SSH client to access the ISLET menu which launches available configurations upon selection. Building on a cross-platform and proven remote access tool like SSH opens the door to greater accessibility that wouldn't otherwise be possible when hypervisors are required, e.g. using smart phones, tablets, and other mobile devices to access training environments.

The three step process to create and deploy a training environment with ISLET is as follows:

1. Have a docker image with the tools needed for training, installed and configured.
2. Create an ISLET configuration file for the image describing its functionality and resources.
3. Place the ISLET configuration file into the /etc/islet directory. After the final step, students can connect to the system and launch the new configuration which will place them into a container based on their image configuration of choice.

A 64-bit Linux operating system is required to run Docker. The recommended operating system for ISLET is Ubuntu, and installation plus configuration for this operating system is very simple with the following make targets:

```
$ sudo apt-get install make
➥ sqlite
$ git clone https://github.com/
➥jonschipp/islet
$ cd islet
$ sudo make install
$ sudo make install-docker
$ sudo make user-config
$ sudo make security-config
$ sudo make install-sample-nsm #
➥Install a few sample config files
```

You can then use ISLET by ssh'ing to the system with a user account and password of demo. Hundreds of training environments for different pieces of software can be made available on an ISLET server from which a user can choose to begin work instantly.

Future work includes porting ISLET to FreeBSD by using jails and implementing a distributed setup to handle large participant numbers seen in Massive Open Online Courses.

Use Cases

At BroCon 14, the precursor to ISLET was introduced to aid in teaching the Bro programming language to participants. The ISLET system ran on Amazon EC2 as an m3.xlarge instance and handled 50+ users simultaneously without issue. The University of Illinois at Urbana-Champaign is using ISLET in their Digital Forensics II course to teach Volatility, Bro, The Sleuthkit, SIFT Kit, and BitCurator. The Open Network Security Monitoring group (OpenNSM) has used ISLET to teach OSSEC, among other tools, and had its first case where a student followed along on their smart phone via an SSH application. The UIUC Linux User Group uses ISLET to teach a C programming series each week, in addition to other Linux tools.

Try It Out

If you would like to try out ISLET, I have two publicly available servers (demo:demo) for experimenting and a vagrant box[3]:

```
$ ssh demo@islet1.jonschipp.com
$ ssh demo@islet2.jonschipp.com
```

References & More Information:

[1] http://www.haifux.org/lect
➥ures/320/netLec8_final.pdf - Linux Containers and the Future Cloud

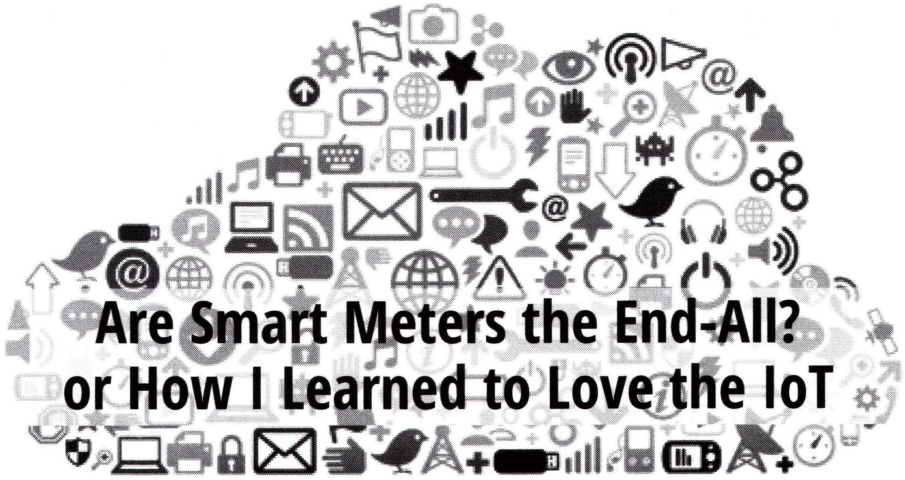
[2] http://domino.research.ibm
➥.com/library/cyberdig.nsf/
➥papers/0929052195DD819C85257D
➥2300681E7B/$File/rc25482.pdf - An Updated Performance Comparison of Virtual Machines and Linux Containers

[3] http://github.com/jonschipp/
➥vagrant/tree/master/islet - Vagrant box

http://github.com/jonschipp/
➥islet - ISLET Source

http://jonschipp.com/talks/
➥ISLET.pdf - Hack3rcon 5 ISLET Presentation

https://www.youtube.com/watch
➥?v=U0KFrsB6f0Q - Hack3rcon 5 ISLET Video



Are Smart Meters the End-All? or How I Learned to Love the IoT

by lg0p89

It seems as though technology continually is making further inroads into our lives, step by step. Beginning from the 1980s to our current time, the computer in theory was supposed to make our existence easier and less stressful. There was supposed to be a four day work week, as we would become so efficient, that is all we would need to get all of the work done. What this did instead was make us work more, as we could get more work done now. We were supposed to be paperless, as we could scan everything into the system, so there would not be a need for paper to be used. Here, the attorneys got involved, stating that we could not prove the digital signature was used by the person alleged to have used the signature, and the paper could not go to the wayside.

On other fronts, technology has paid off for the users. All we have to do is look to our smart phone, USB, and improved satellites. If we want to know exactly where we are standing on the planet, there is an app for that.

The current hardware innovations are part of the Internet of Things (IoT). An entertaining example of this as of late are the light bulbs that flash different colors based on the user's parameters. For instance, if the Detroit Lions win their football game, this may be set to flash blue. This is in the pipeline to becoming yet more mainstream.

The latest iteration involves energy meters. We are all familiar with the usual meter on the side of the house or on the apartment building with the glass globe (looking like the robot on its side from *Lost in Space*), with the dials spinning. In the late summer, the spinning dials appear to be moving near the speed of light as our air conditioners are running.

Hardware

Visually and physically, the meter is still in a roundish housing and attached to the dwelling. The prior meters required the energy worker to walk around once a month and read the numbers. The new smart meters are digital and automatically gather the data and transmit this to the electric company. There would be no person walking around your property leaving footprints in the snow. The structure and housing itself has a few variants in size, but these are mostly the same. The fundamental, pertinent change involves the software. This allows the meter to communicate and send various data to the electric company.

Benefits

Clearly, there is a benefit to someone or some entity. Otherwise, the expense would not have been incurred for the R&D. Big Business does not need to spend the funds on a shiny new toy. The primary beneficiary is the electric company. They no longer have to employ the people to walk around and read the meters for everyone. This cuts the expenses for labor and the associated overhead. The new meters monitor the usage per household, and subsequently per neighborhood, county, etc. Big data would be so happy with this. With the quicker monitoring comes the ability to react to geographic power needs faster. If an area has a higher need for power, the situation can be monitored so other areas where the consumption is less can have this transferred. This is much like loan balancing on the servers.

Not So Beneficial

This prima facie sounds great. Allegedly, this benefits the electric company and society, and the electricity should be less costly. But everything comes at a cost. There still is no free lunch. All of the attributes are not good for the consumers. A primary concern has been its security and privacy. The hardware itself is hackable with little effort. This is unnerving, at best. Also, the information on the personal usage for the residence and, by extension, other information can be accessed by others. A thief could monitor your usage and, if it appears the usage is well below the baseline for two or three days, could believe you are on vacation and break into your residence.

Hackable

The hardware is attached to your home, condo, duplex, apartment building on the outside of the structure. Anyone could simply walk up and look at the different access points to hack on the hardware. If it is during the day during the work week, no one would probably even notice. If someone were to walk up to this person, it would not be that difficult for them to social engineer their way out of this. The trespasser could not only get the raw usage data, but also any other data the hardware holds (e.g. account number).

This sounds a bit far-fetched. It does not seem likely that a piece of equipment that records your electrical usage would be that much of a detriment. Well, it happened. Beginning in 2009, there were power thefts throughout Puerto Rico. This became a significant issue and the FBI began investigating the thefts. The FBI believed this was due to the "new and improved" smart meters being deployed. It appeared from the investigation that people previously employed by the company that manufactured the meters, along with current employees of the utility company, were involved with the theft.

The people were charging $300-$1,000 for residential customers and $3,000 for the commercial meters for the unlawful services. Some of the estimates concluded the utility company lost millions in revenue due to this. This was done by using an optical converter device attached to a laptop and software downloaded from the Internet. There are several tools that can do this. One open source tool is the Termineter. This also uses the optical interface as the access point. The hardware for this costs $300-$400. To fully implement this does not take a significant capital outlay. In essence, the tool merely changes the ratio of how the meter records the electricity used.

The person did not have to open the meter, cut the metal band, or anything physical. They just had to walk over to it with their laptop and an optical converter device. It wasn't complicated or even a two-step process.

In Short...

Overall, technology is our friend. It may give us a temporary headache but, in the long run, it makes our life easier. The smart meter is one such item. It makes sense to use it. The more data the electric company has access to, the better they can plan for the usage. This improves their operations, which translates into electrical savings for the consumer. With the good comes the bad. The software written to manipulate the smart meter was coded more with the focus being on how to operate and record the electrical usage versus security. The level of security has already proven to be financially disastrous for at least one utility. With the promulgation of open source software and the relatively low cost of the hardware to hack the smart meter, there will be issues until there are patches written to rid the system openings that anyone can get into.

For Further Thoughts

Geib, A. How privacy-conscious consumers are fooling, hacking smart meters. http://www.naturalnews.com/036476_smart_meters_hacking_privacy.html

Kumar, M. Open source smart meter hacking framework can hack into the power grid. http://thehackernews.com/2012/07/open-source-smart-meter-hacking.html

Protalinski, E. Smart meter hacking tool released. http://www.zdnet.com/smart-meter-hacking-tool-released-7000001338/

Sunshine, W.L. Pros and cons of smart meters. http://energy.about.comp/od/metering/a/Pros-And-Cons-of-Smart-Meters.htm

Tweed, K. FBI finds smart meter hacking surprisingly easy. http://www.greentechmedia.com/articles/read/fbi-finds-smart-meter-hacking-surprisingly-easy

HACKER HAPPENINGS

Listed here are some upcoming events of interest to hackers. Hacker conferences generally cost under $150 and are open to everyone. Higher prices may apply to the more elaborate events such as outdoor camps. If you know of a conference or event that should be known to the hacker community, ***email us*** at **happenings@2600.com** or by snail mail at **Hacker Happenings, PO Box 99, Middle Island, NY 11953 USA.** We only list events that have a firm date and location, aren't ridiculously expensive, are open to everyone, and welcome the hacker community.

April 25-26
Maker Faire U.K.
Life Science Centre
Newcastle upon Tyne, England
www.makerfaireuk.com

May 14-15
THOTCON 0x6
Chicago, Illinois
thotcon.org

May 16-17
Maker Faire Bay Area
San Mateo Event Center
San Mateo, California
www.makerfaire.com

June 3-5
RVAsec
Virginia Commonwealth University
Richmond, Virginia
rvasec.com

June 12-14
CircleCityCon
Westin Indianapolis
Indianapolis, Indiana
circlecitycon.com

June 12-14
NolaCon
Crowne Plaza New Orleans
New Orleans, Louisiana
nolacon.com

June 20-21
Nuit Du Hack
Académie Fratellini
Paris, France
www.nuitduhack.com

June 27-28
Maker Faire Kansas City
Union Station
Kansas City, Missouri
www.makerfaire.com

July 25-26
Maker Faire Detroit
The Henry Ford
Dearborn, Michigan
www.makerfaire.com

August 6-9
DEF CON 23
Paris/Bally's
Las Vegas, Nevada
www.defcon.org

August 13-17
Chaos Communication Camp
Ziegeleipark Mildenberg
Zehdenick, Germany
www.ccc.de

September 23-27
DerbyCon
Hyatt Regency
Louisville, Kentucky
www.derbycon.com

September 26-27
World Maker Faire New York
New York Hall of Science
Queens, New York
www.makerfaire.com

October 9-10
GrrCON
DeVos Place
Grand Rapids, Michigan
www.grrcon.org

December 27-30
Chaos Communication Congress
Congress Center Hamburg
Hamburg, Germany
www.ccc.de

Please send us your feedback on any events you attend and let us know if they should/should not be listed here.

Marketplace

For Sale

CLUB-MATE is now easy to get in the United States! The caffeinated German beverage is a huge hit at any hacker gathering. Available in two quantities: $36.99 per 12 pack or $53.99 per 18 pack of half liter bottles plus shipping. [Check to see if we have any of the limited Winter Edition still in stock!] Write to contact@club-mate.us or order directly from store.2600.com. We are now working to supply stores nationwide - full details at club-mate.us.

HAXWEAR.COM sells authentic handmade t-shirts for hackers, by hackers. While you're hacking the Gibson there's no reason you can't look awesome at the same time rocking one of our exclusive designs! Every order gets a free gift. Check us out at www.haxwear.com

HOME NETWORK SECURITY APPLIANCE blocks exploits, malware, and CnC traffic. Powerful, affordable, and hacker friendly device runs open-source software including OpenWrt, Snort, Squid, ClamAV, and more. Kickstarter funded hardware runs enterprise-grade network security processors in small form-factor fanless platform. Order online from ITUSnetworks.com

PORTABLE PENETRATOR. Crack WEP, WPA, WPA2 Wi-Fi networks. Coupon code for Portable Penetrator with Cracking Suite. Get 20% off with coupon code 2600 at http://shop.secpoint.com/2600

GAMBLING MACHINE JACKPOTTERS, portable magnetic stripe readers & writers, RFID reader writers, lockpicks, vending machine jackpotters, concealable blackjack card counting computers, poker cheating equipment, computer devices, odometer programmers, and much more. www.hackershomepage.com

OPEN POWER: *Electoral Reform Act of 2015 - Open Source Activist Tool Kit* by HOPE speaker Robert Steele available on the Kindle and at amazon.com

HACKER WAREHOUSE is your one stop shop for hacking equipment. We understand the importance of tools and gear which is why we strive to carry only the highest quality gear from the best brands in the industry. From WiFi Hacking to Hardware Hacking to Lock Picks, we carry equipment that all hackers need. Check us out at HackerWarehouse.com.

HACKERSTICKERS.COM sells great hacker, programmer, and security gear such as shirts, caffeinated candy, laptop stickers, and lock pick sets. Get a free sticker with purchase, just add to cart and enter "freestick" at checkout.

CAPT'N CRUNCH WHISTLE. Only a few left. THIS IS *THE ORIGINAL* WHISTLE from Capt'n Crunch cereal box. Brand new, unused, mint condition! Join the elite few who own this treasure! Once the remaining few are sold, that's it - there will never, *ever,* be another one offered again. Key chain hole for easy insertion on your key ring. Identify yourself at meetings, etc. as a *2600* member by dangling your key chain and saying nothing. Cover one hole and produce *exactly* 2600 hz. Cover the other hole and you get another frequency. Use both holes to call your dog, dolphin, concubine, or hamster. Also, ideal for telephone remote control of your own electronic remote devices. Price includes mailing: $59.95. Not only a rare collector's item but a VERY USEFUL and unique device which is easy to carry with you at all times; nobody will ever know, except you, how it is used for remote control! Cash/money order only. Mail to: WHISTLE, P.O. Box 28992 (ST); CC, Missouri 63132.

A TOOL TO TALK TO CHIPS. It's the middle of the night. You compile and program test code for what must be the 1000th time. Digging through the datasheets again, you wonder if the problem is in your code, a broken microcontroller... who knows? There are a million possibilities, and you've already tried everything twice. Imagine if you could take the frustration out of learning about a new chip. Type a few intuitive commands into the Bus Pirate's simple console interface. The Bus Pirate translates the commands into the correct signals, sends them to the chip, and the reply appears on the screen. No more worry about incorrect code and peripheral configuration, just pure development fun for only $30 including world wide shipping. Check out this open source project and more at DangerousPrototypes.com.

Announcements

JOIN THE MOVEMENT! Help us expose the Justice Department's political agenda against hackers! We are blowing the Ghost Exodus case wide open and exposing the perpetrators responsible for manufacturing and slanting his case in favor of the prosecution, ironically, the same prosecutor residing over the case of Barrett Brown and Matthew Weigman. Find out why Jesse McGraw's lawyer refuses to file his appeal, and what one rogue prosecutor is trying to cover up. Help us to distribute pamphlets at hacker conferences and visit our legal fund to donate to the cause. Free Ghost Exodus! Free Jesse! Fundraiser: http://tinyurl.com/freeghostexodus Contact: freejesselegalteam@hush.ai Main Site (still under construction): http://freejesselegal.wix.com/freejesse

Wanted

WE ARE AN UNDERGROUND EXPERIMENTAL DUBSTEP RAP BAND along the lines of the Beastie Boys and Mindless Self Indulgence, creating music outside the system exclusively for the Internet. We are in need of an awesome web designer to redesign our outdated wordpress website: www.tvmessiah.com. Check out our latest tracks on youtube (http://www.youtube.com/user/tvmessiah/videos) and, if you dig us and believe we are worthy, please reach out to us: number7@tvmessiah.com.

Services

LISTEN TO THE SYNACK PACK PODCAST. There are many security minded podcasts out there, and we're one of them. We are here for the newbies and veterans alike! The SYNACK Pack podcast discusses general news as well as technology specific issues, all from a hacker perspective. Have a listen and we LOVE feedback! http://synackpack.com

SUSPECTED OR ACCUSED OF INTERNET-RELATED CRIMINAL OFFENSES? Stand up for your rights! Be calm, respectful, and clear: "I respectfully invoke all of my Constitutional rights, officer. I do not consent to any search or seizure, I choose to remain silent, and I want to speak to a lawyer." Remember basic game theory and the Prisoner's Dilemma: if nobody talks,

then everybody walks. In the event of unwanted police contact, it would be advisable to consult with a lawyer experienced in defending human beings facing computer-related accusations in California and federal courts. I am an aggressive Constitutional and criminal defense lawyer with experience representing persons accused of hacking, cracking, misappropriation of trade secrets, and other cybercrimes. I am a semantic warrior committed to the liberation of information (after all, information wants to be free and so do we), and I am willing to contribute pro bono representation for whistleblowers and accused hackers acting without malice. Past clients include Kevin Mitnick (million-dollar-bail case in California Superior Court dismissed), Robert Lyttle of The Deceptive Duo (patriotic hacktivist who exposed elementary vulnerabilities in the United States information infrastructure), and Vincent Kershaw, reported member of Anonymous indicted for his alleged participation in a DDOS action against PayPal. Also, given that the worlds of the hacker and the cannabis connoisseur have often intersected historically, please note I also specialize in Constitutional and criminal defense of cannabis cases as well as legal compliance with a complex maze of marijuana-related laws and regulations. Please contact Omar Figueroa at (415) 489-0420 or (707) 829-0215, at omar@stanfordalumni.org, or at Law Offices of Omar Figueroa, 7770 Healdsburg Ave., Ste. A, Sebastopol, CA 95472.

DIGITAL FORENSICS FOR THE DEFENSE! Sensei Enterprises believes in the Constitutional right to a zealous defense, and backs up that belief by providing the highest quality digital forensics and electronic evidence support for criminal defense attorneys. Our veteran experts are cool under fire in a courtroom - and their forensic skills are impeccable. We recover data from many sources, including computers, external media, and smartphones. We handle a wide range of cases, including hacking, child pornography possession/distribution, solicitation of minors, theft of proprietary data, data breaches, interception of electronic communications, identity theft, rape, murder, embezzlement, wire fraud, racketeering, espionage, cyber harassment, cyber abuse, terrorism, and more. Sensei's digital forensic examiners all hold prestigious forensic certifications. Our principals are co-authors of *The Electronic Evidence Handbook* (American Bar Association 2006) and hundreds of articles on digital forensics and electronic evidence. They lecture throughout North America and have been interviewed by ABC, NBC, CBS, CNN, Reuters, many newspapers, and even Oprah Winfrey's *O* magazine. For more information, call us at 703-359-0700 or email us at sensei@senseient.com.

INTELLIGENT HACKERS UNIX SHELL: Reverse.Net is owned and operated by intelligent hackers. We believe every user has the right to online security and privacy. In today's hostile anti-hacker atmosphere, intelligent hackers require the need for a secure place to work, compile, and explore without big-brother looking over their shoulder. Hosted in Chicago with Filtered DoS Protection. Multiple Dual Core FreeBSD servers. Affordable pricing from $5/month, with a money back guarantee. Lifetime 26% discount for *2600* readers. Coupon Code: Save2600. http://www.reverse.net/

GET YOUR HAM RADIO LICENSE! KB6NU's "No-Nonsense" Study Guides make it easy to get your Technician Class or General Class amateur radio license. They clearly and succinctly explain the concepts, while at the same time give you the answers to all of the questions on the test. And the best part is that they are free from www.kb6nu.com/tech-manual. E-mail cwgeek@kb6nu.com for more information.

SECURE UNIX SHELLS & HOSTING SINCE 1999. JEAH.NET is one of the oldest and most trusted for fast, stable shell accounts. We provide hundreds of vhost domains for IRC and email, the latest popular *nix programs, access to classic shell programs and compilers. JEAH.NET proudly hosts eggdrop, BNC, IRCD, and web sites w/SQL. *2600* readers' setup fees are always waived. BTW: FYNE.COM (our sister co.) offers free DNS hosting and WHOIS privacy for $3.50 with all domains registered or transferred in!

Personal

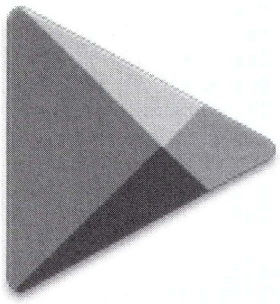
I AM CURRENTLY LOCKED UP in federal prison and would love to have a pen-pal or three to write. My interests include social engineering, politics, and journalism. If possible, I'm also looking for book or magazine donations. Mag and paperback donations can be sent by private parties, but hardbacks must be sent from the publisher or bookstore. My release is in 2020, so I'd really like to keep up on all the changes going on, as well as talk to like-minded people regarding any topic computer-related or politics and S.E. Thanks and always keep "HOPE"-ing for a better life out there. Write to: Anthony B. Ellrodt #65321-097, FCC Beaumont - Low, P.O. Box 26020, Beaumont, TX 77720-6020.

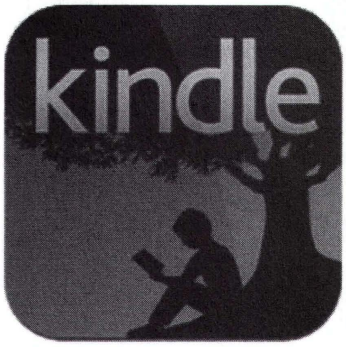
NO ONE WILL BE AT MY FUNERAL. I am requesting an English speaking/writing international pen pal. I am male living in the U.S. with interests ranging from radio, hacking, satellite technology (outer space), military tech/communication/intel, photography, and beyond. Send me a Euro dollar or your region's currency, for entertainment purposes only. I am enlightened by your intelligence and capabilities. Respond via snail mail to KW, PO Box 61, Burleson, Texas 76097.

BEING CLOSE TO RELEASE IN 2016, I am looking to brush up on what's been going on in the hacker world. I would be interested in discussing topics, getting articles mailed in, or book recommendations (or donations). Some topics I am familiar with include SQL, PHP, Wi-Fi, and pen testing. I am also interested in any info anyone will provide about speaking topics at events like Defcon or HOPE. I've been locked up since 2009 so any info, articles, or speaking topics anyone wants to send, or anyone just wanting to chat with me, would be greatly appreciated. I can be reached through Jpay.com using my DoC #339317 in Washington State or via mail at Chris Berge, 339317 10-G31, Washington State Penitentiary, 1313 N 13th Ave., Walla Walla, WA 99362. Please note that book donations must come from a company and have a receipt. Happy hacking!

ONLY SUBSCRIBERS CAN ADVERTISE IN *2600!* Don't even think about trying to take out an ad unless you subscribe! All ads are free and there is no amount of money we will accept for a non-subscriber ad. We hope that's clear. Of course, we reserve the right to pass judgment on your ad and not print it if it's amazingly stupid or has nothing at all to do with the hacker world. We make no guarantee as to the honesty, righteousness, sanity, etc. of the people advertising here. Contact them at your peril. All submissions are for ONE ISSUE ONLY! If you want to run your ad more than once you must resubmit it each time. Don't expect us to run more than one ad for you in a single issue either. Include your address label/envelope or a photocopy so we know you're a subscriber. If you're an electronic subscriber, please send us a copy of your subscription receipt. Send your ad to *2600* Marketplace, PO Box 99, Middle Island, NY 11953. You can also email your ads to subs@2600.com.

Deadline for Summer issue: 5/21/15.





Have you seen a digital copy of *2600*? In addition to the good old-fashioned paper version, you can now subscribe in more parts of the world than ever via Google Play and the Kindle. We're also constantly increasing our library of back issues and Hacker Digests.

Head to digital.2600.com for the latest

Did you miss HOPE X?

Or were you there and now you miss it because it's over? Either way, we're here to help.

We have HOPE X leftover shirts with the snazzy HOPE X badge design in the front and the colorful artwork on the back, all on a charcoal gray colored shirt. $20 each while supplies last - store.2600.com/shirts.html. Did you somehow manage to miss one of the 100 talks that were presented? DVDs of ALL of the three speaker tracks are available for only $5 each, $399 for all 102 DVDs. We can't possibly print all of the talk titles here, but you can see them at store.2600.com/hopex2014.html and select the ones you want. And for the first time ever, we're offering all of the talks on flash drives (either two 32gb or one 64gb drive). Much higher quality than what's online, no DRM, easy to copy, sharing encouraged. Only $99 for the entire set at store.2600.com/hofldr.html

New! We now have 64gb flash drives containing ALL of the talks from three more of our conferences for only $69 each! (The Last HOPE, The Next HOPE, and HOPE Number Nine)
Look for details on our store.

"Journalism is printing what someone else does not want printed. Everything else is public relations" - George Orwell

Editor-In-Chief
Emmanuel Goldstein

Associate Editor
Bob Hardy

Layout and Design
Skram

Cover
Dabu Ch'wald

Office Manager
Tampruf

STAFF

Infrastructure
flyko

Network Operations
phiber

Broadcast Coordinator
Juintz

IRC Admins
beave, koz, r0d3nt

Inspirational Music: Arab Strap, Command & Conquer Red Alert, Ladytron, Faust and Shortee, the 13th Floor Elevators

Shout Outs: Pope jonnyX, ProgressBar, brmlab, c3d2, Radiofabrik, Chaostreff Salzburg

***2600* is written by members of the global hacker community. You can be a part of this by sending your submissions to articles@2600.com or the postal address below.**

2600 *(ISSN 0749-3851, USPS # 003-176); Spring 2015, Volume 32 Issue 1, is published quarterly by 2600 Enterprises Inc., 2 Flowerfield, St. James, NY 11780. Periodical postage rates paid at St. James, NY and additional mailing offices.*

POSTMASTER:
Send address changes to: *2600*
P.O. Box 752 Middle Island,
NY 11953-0752.

SUBSCRIPTION CORRESPONDENCE:
2600 Subscription Dept., P.O. Box 752,
Middle Island, NY 11953-0752 USA
(subs@2600.com)

YEARLY SUBSCRIPTIONS:
U.S. & Canada - $27 individual,
$50 corporate (U.S. Funds)
Overseas - $38 individual, $65 corporate

BACK ISSUES:
1984-1999 are $25 per year when available.
Individual issues for 1988-1999
are $6.25 each when available.
2000-2014 are $27 per year or $6.95 each.
Shipping added to overseas orders.

LETTERS AND ARTICLE SUBMISSIONS:
2600 Editorial Dept., P.O. Box 99,
Middle Island, NY 11953-0099 USA
(letters@2600.com, articles@2600.com)

***2600* Office/Fax Line: +1 631 751 2600**

Copyright © 2015; 2600 Enterprises Inc.

MEETINGS

ARGENTINA

Buenos Aires: Bodegon Bellagamba, Carlos Calvo 614, San Telmo. In the back tables passing bathrooms.

AUSTRALIA

Central Coast: Ourimbah RSL (in the TAB area), 6/22 Pacific Hwy. 6 pm

Melbourne: Oxford Scholar Hotel, 427 Swanston St.

Sydney: The Crystal Palace Hotel, 789 George St. 6 pm

AUSTRIA

Graz: Cafe Haltestelle on Jakominiplatz.

BELGIUM

Antwerp: Central Station, top of the stairs in the main hall. 7 pm

BRAZIL

Belo Horizonte: Pelego's Bar at Assufeng, near the payphone. 6 pm

CANADA

Alberta

Calgary: Food court of Eau Claire Market. 6 pm

Edmonton: Elephant & Castle Pub, 10314 Whyte Ave, near big red telephone box. 6 pm

British Columbia

Kamloops: Student St in Old Main in front of Tim Horton's, TRU campus.

Vancouver (Surrey): Central City Shopping Centre food court by Orange Julius.

Manitoba

Winnipeg: St. Vital Shopping Centre, food court by HMV.

New Brunswick

Moncton: Champlain Mall food court, near KFC. 7 pm

Newfoundland

St. John's: Memorial University Center food court (in front of the Dairy Queen).

Ontario

Ottawa: World Exchange Plaza, 111 Albert St, second floor. 6:30 pm

Toronto: Free Times Cafe, College and Spadina.

Windsor: Sandy's, 7120 Wyandotte St E. 6 pm

CHINA

Hong Kong: Pacific Coffee in Festival Walk, Kowloon Tong. 7 pm

COSTA RICA

Heredia: Food court, Paseo de las Flores Mall.

CZECH REPUBLIC

Prague: Legenda pub. 6 pm

DENMARK

Aalborg: Fast Eddie's pool hall.

Aarhus: In the far corner of the DSB cafe in the railway station.

Copenhagen: Cafe Blasen.

Sonderborg: Cafe Druen. 7:30 pm

FINLAND

Helsinki: Fenniakortteli food court (Vuorikatu 14).

FRANCE

Cannes: Palais des Festivals & des Congres la Croisette on the left side.

Grenoble: EVE performance hall on the campus of Saint Martin d'Heres. 6 pm

Lille: Grand-Place (Place Charles de Gaulle) in front of the Furet du Nord bookstore. 7:30 pm

Paris: Cafe Monde et Medias, Place de la Republique. 6 pm

Rennes: Bar le Golden Gate, Rue St Georges a Rennes. 8 pm

Rouen: Place de la Cathedrale, benches to the right. 8 pm

Toulouse: Place du Capitole by the benches near the fast food and the Capitole wall. 7:30 pm

GREECE

Athens: Outside the bookstore Papasotiriou on the corner of Patision and Stournari. 7 pm

IRELAND

Dublin: At the payphones beside the Dublin Tourism Information Centre on Suffolk St. 7 pm

Westport: Phone booth next to the library. 7 pm

ISRAEL

***Beit Shemesh:** In the big Fashion Mall (across from train station), second floor, food court. Phone: 1-800-800-515. 7 pm

***Safed:** Courtyard of Ashkenazi Ari.

ITALY

Milan: Piazza Loreto in front of McDonalds.

JAPAN

Kagoshima: Amu Plaza next to the central railway station in the basement food court (Food Cube) near Doutor Coffee.

Tokyo: Mixing Bar near Shinjuku Station, 2 blocks east of east exit. 6:30 pm

MEXICO

Chetumal: Food court at La Plaza de Americas, right front near Italian food.

Mexico City: "Zocalo" Subway Station (Line 2 of the "METRO" subway, the blue one). At the "Departamento del Distrito Federal" exit, near the payphones and the candy shop, at the beginning of the "Zocalo-Pino Suarez" tunnel.

NETHERLANDS

Utrecht: In front of the Burger King at Utrecht Central Station. 7 pm

NORWAY

Oslo: Sentral Train Station at the "meeting point" area in the main hall. 7 pm

Tromsoe: The upper floor at Blaa Rock Cafe, Strandgata 14. 6 pm

PERU

Lima: Barbilonia (ex Apu Bar), en Alcanfores 455, Miraflores, at the end of Tarata St. 8 pm

Trujillo: Starbucks, Mall Aventura Plaza. 6 pm

PHILIPPINES

Quezon City: Chocolate Kiss ground floor, Bahay ng Alumni, University of the Philippines Diliman. 4 pm

SWEDEN

Stockholm: Starbucks at Stockholm Central Station.

SWITZERLAND

Lausanne: In front of the MacDo beside the train station. 7 pm

UNITED KINGDOM

England

Brighton: At the phone boxes by the Sealife Centre (across the road from the Palace Pier). Payphone: (01273) 606674. 7 pm

Leeds: The Brewery Tap Leeds. 7 pm

London: Trocadero Shopping Center (near Piccadilly Circus), lowest level. 6:30 pm

Manchester: Bulls Head Pub on London Rd. 7:30 pm

Norwich: Entrance to Chapelfield Mall, under the big screen TV. 6 pm

Wales

Ewloe: St. David's Hotel.

UNITED STATES

Alabama

Auburn: The student lounge upstairs in the Foy Union Building. 7 pm

Huntsville: Upstairs at Tenders, 800 Holmes Ave NE. 6 pm

Arizona

Phoenix: HeatSync Labs, 140 W Main St. 6 pm

Prescott: Method Coffee, 3180 Willow Creek Rd. 6 pm

Arkansas

Ft. Smith: River City Deli at 7320 Rogers Ave. 6 pm

California

Los Angeles: Union Station, inside main entrance (Alameda St side) between Union Bagel and the Traxx Bar.

Monterey: East Village Coffee Lounge. 5:30 pm

Orange: Orange Circle. 7 pm

Sacramento: Hacker Lab, 1715 I St.

San Diego: Regents Pizza, 4150 Regents Park Row #170.

San Francisco: 4 Embarcadero Center near street level fountains. 6 pm

San Jose: Outside the cafe at the MLK Library at 4th and E San Fernando. 6 pm

Colorado

Loveland: Starbucks at Centerra (next to Bonefish Grill). 7 pm

Connecticut

Newington: Panera Bread, 3120 Berlin Tpke. 6 pm

Delaware

Newark: Barnes and Nobles cafe area, Christiana Mall.

District of Columbia

Arlington: Rock Bottom at Ballston Commons Mall. 7 pm

Florida

Fort Lauderdale: Undergrounds Coffeehaus, 3020 N Federal Hwy. 7 pm

Gainesville: In the back of the University of Florida's Reitz Union food court. 6 pm

Jacksonville: O'Brothers Irish Pub, 1521 Margaret St. 6:30 pm

Melbourne: Sun Shoppe Cafe, 540 E New Haven Ave. 5:30 pm

Sebring: Lakeshore Mall food court, next to payphones. 6 pm

Titusville: Krystal Hamburgers, 2914 S Washington Ave (US-1).

Georgia

Atlanta: Lenox Mall food court. 7 pm

Hawaii

Hilo: Prince Kuhio Plaza food court, 111 East Puainako St.

Idaho

Boise: BSU Student Union Building, upstairs from the main entrance. Payphones: (208) 342-9700.

Pocatello: Flipside Lounge, 117 S Main St. 6 pm

Illinois

Chicago: Golden Apple, 2971 N. Lincoln Ave. 6 pm

Peoria: Starbucks, 1200 West Main St.

Indiana

Evansville: Barnes & Noble cafe at 624 S Green River Rd.

Indianapolis: Tomlinson Tap Room in City Market, 222 E Market St.

Iowa

Ames: Memorial Union Building food court at the Iowa State University.

Davenport: Co-Lab, 1033 E 53rd St.

Kansas

Kansas City (Overland Park): Barnes & Noble cafe, Oak Park Mall.

Wichita: Riverside Perk, 1144 Bitting Ave.

Louisiana

New Orleans: Z'otz Coffee House uptown, 8210 Oak St. 6 pm

Maine

Portland: Maine Mall by the bench at the food court door. 6 pm

Maryland

Baltimore: Barnes & Noble cafe at the Inner Harbor.

Massachusetts

Boston: Stratton Student Center (Building W20) at MIT in the 2nd floor lounge area. 7 pm

Worcester: TESLA space - 97D Webster St.

Michigan

Ann Arbor: Starbucks in The Galleria on S University. 7 pm

Minnesota

Bloomington: Mall of America food court in front of Burger King. 6 pm

Missouri

St. Louis: Arch Reactor Hacker Space, 2400 S Jefferson Ave.

Montana

Helena: Hall beside OX at Lundy Center.

Nebraska

Omaha: Westroads Mall food court near south entrance, 100th and Dodge. 7 pm

Nevada

Elko: Uber Games and Technology, 1071 Idaho St. 6 pm

Las Vegas: SYN Shop, 117 N 4th St. 7 pm

Reno: Barnes & Noble Starbucks 5555 S. Virginia St.

New Hampshire

Keene: Local Burger, 82 Main St. 7 pm

New Jersey

Morristown: Panera Bread, 66 Morris St. 7 pm

Somerville: Dragonfly Cafe, 14 E Main St.

New York

Albany: Starbucks, 1244 Western Ave. 6 pm

New York: Citigroup Center, in the lobby, 153 E 53rd St, between Lexington & 3rd.

Rochester: Interlock Rochester, 1115 E Main St, Door #7, Suite 200. 7 pm

North Carolina

Charlotte: Panera Bread, 9321 JW Clay Blvd (near UNC Charlotte). 6:30 pm

Greensboro: Caribou Coffee, 3109 Northline Ave (Friendly Center).

Raleigh: Cup A Joe, 3100 Hillsborough St. 7 pm

North Dakota

Fargo: West Acres Mall food court.

Ohio

Cincinnati: Hive13, 2929 Spring Grove Ave. 7 pm

Cleveland (Warrensville Heights): Panera Bread, 4103 Richmond Rd. 7 pm

Columbus: Front of the food court fountain in Easton Mall. 7 pm

Dayton: Marions Piazza ver. 2.0, 8991 Kingsridge Dr., behind the Dayton Mall off SR-741.

Youngstown (Niles): Panara Bread, 5675 Youngstown Warren Rd.

Oklahoma

Oklahoma City: Cafe Bella, southeast corner of SW 89th St and Penn.

Oregon

Portland: Theo's, 121 NW 5th Ave. 7 pm

Pennsylvania

Allentown: Panera Bread, 3100 W Tilghman St. 6 pm

Harrisburg: Panera Bread, 4263 Union Deposit Rd. 6 pm

Philadelphia: 30th St Station, food court outside Taco Bell.

Pittsburgh: Tazz D'Oro, 1125 North Highland Ave at round table by front window.

State College: in the HUB above the Sushi place on the Penn State campus.

Puerto Rico

San Juan: Plaza Las Americas on first floor.

Trujillo Alto: The Office Irish Pub. 7:30 pm

South Dakota

Sioux Falls: Empire Mall, by Burger King.

Tennessee

Knoxville: West Town Mall food court. 6 pm

Memphis: Republic Coffee, 2924 Walnut Grove Rd. 6 pm

Nashville: J&J's Market & Cafe, 1912 Broadway. 6 pm

Texas

Austin: Spider House Cafe, 2908 Fruth St, front room across from the bar. 7 pm

Dallas: Wild Turkey, 2470 Walnut Hill Ln. 7 pm

Houston: Ninfa's Express seating area, Galleria IV. 6 pm

Plano: Fourteen Eighteen Coffeehouse, 1418 Ave K. 6 pm

Vermont

Burlington: The Burlington Town Center Mall food court under the stairs.

Virginia

Arlington: (see District of Columbia)

Blacksburg: Squires Student Center at Virginia Tech, 118 N. Main St. 7 pm

Charlottesville: Panera Bread at the Barracks Road Shopping Center. 6:30 pm

Richmond: Hack.RVA 1600 Roseneath Rd. 6 pm

Washington

Seattle: Washington State Convention Center. 2nd level, south side. 6 pm

Spokane: The Service Station, 9315 N Nevada (North Spokane).

Wisconsin

Madison: Fair Trade Coffee House, 418 State St.

All meetings take place on the first Friday of the month (a * indicates a meeting that's held on the first Thursday of the month). Unless otherwise noted, *2600* meetings begin at 5 pm local time. To start a meeting in your city, send email to meetings@2600.com.

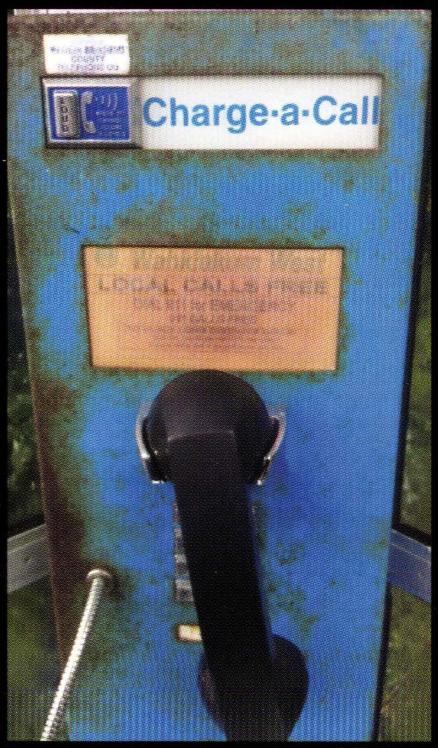
Free Payphones

France. This free phone was found at the bottom of Mont Salève. It looks as if its dialing options are a bit limited.

Photo by Jonathan Dumont

Austria. Ski resorts are apparently a popular spot for free phones. This one in Nassfeld is programmed to dial a local taxi. However, it can be defeated with touch tones through the mouthpiece.

Photo by Richard Hanisch



United States. Now this is a great service (free local calls) offered at this payphone in Rosburg, Washington by this friendly rural phone company.

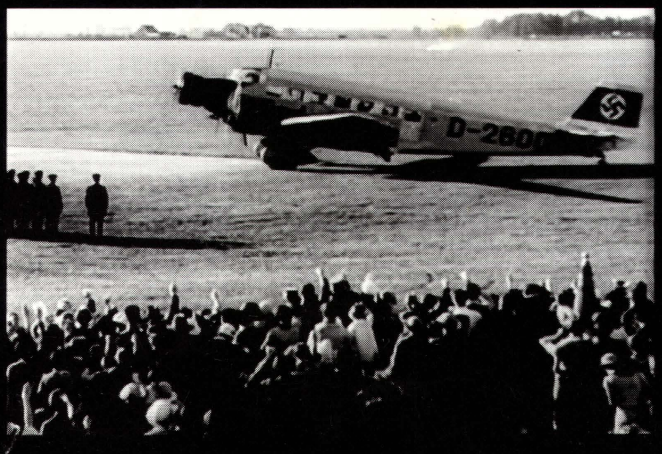
Photo by ZombieRaccoon

Switzerland. Technically, this is a free payphone, since you can make calls without paying, but you would really be annoying the people who run the backpacker hotel it's a part of.

Photo by Nicolas RUFF

Visit **http://www.2600.com/phones/** to see our foreign payphone photos!
(Or turn to the inside front cover to see more right now.)

The Back Cover Photos



Above: Yet another proud-looking building worthy of bearing our name. Seen by **Nodechomsky** in Memphis, Tennessee, this was apparently taken on one of the rare days that our pirate flag wasn't flying.

Left: This image has been sent to us a number of times over the years, so we've finally decided to print it. As noted by **Johannes Grenzfurthner**, this was Hitler's plane, as captured in the 1935 propaganda film *Triumph of the Will*.

If you've spotted something that has "2600" in it or anything else of interest to the hacker world (such as funny uses of "hacker," "unix," "404," you get the idea...), take a picture and send it on in! Be sure to use the highest quality settings on your camera to increase the odds of it getting printed. Make sure and tell us where you spotted your subject along with any other info that makes it interesting - many photos are eliminated due to lack of detail.

Email your submissions to articles@2600.com or use snail mail to *2600* Editorial Dept., PO Box 99, Middle Island, NY 11953 USA.

If we use your picture, you'll get a free one-year subscription (or back issues) or a *2600* t-shirt of your choice.